Soft-ICE Guide de l'utilisateur.

CHAPITRE 1

1.1 Description du produit

Soft-ICE est un outil de déverminage logiciel .

Soft-ICE utilise le mode protégé du 386 pour faire fonctionner DOS dans
une machine virtuelle. Ceci permet à Soft-ICE d'avoir un contrôle total
de l'environnement DOS. Soft-ICE utilise les caractéristiques du mode protégé du 386, telles que le paging, le niveau de privilège des I/O, les point d'arrêts sur registres, pour ajouter des points d'arrêts de niveau hard a votre débugger actuel.

Soft-ICE a été écrit avec trois idées en tête:

1. Utiliser les possibilités de machine virtuelle des 386 pour obtenir des possibilités de débugging impossibles à obtenir ou pour le moins prohibitivement lentes avec un débugger soft.
   (ex: points d'arrêt hard en temps réel, protection mémoire,etc..).
2. Pouvoir travailler avec les débuggers existants. Nous voulions produire un outil fonctionnant avec les produits existants. Nous avons créé Soft-ICE de telle façon que vous ne soyez pas obligé d'apprendre un nouveau débugger pour obtenir de puissantes fonctions de débugging hard.
3. Etre un programme simple d'utilisation, avec des fenêtres qui s'affichent instantanément.
   Toutes les commandes de Soft-ICE ont été créées pour rentrer dans une petite fenêtre de telle sorte que cette information soit toujours visible sur l'écran en-dessous de Soft-ICE. Une aide en ligne dynamique vient en aide aux utilisateurs qui ne se servent de Soft-ICE qu'occasionnellement.

Les caractéristiques du programme Soft-ICE:

* points d'arrêt en temps réel sur les lectures/écritures en mémoire, lectures/écritures de ports, domaines mémoire, et interruptions.
* historique des traces arrières
* débugging niveau source et symbolique
* un environnement fonctionnant avec les débuggers existants
* support complet de l'EMM 4.0
* une fenêtre pouvant être appelée à tout instant
* la possibilité de sortir par un appui touche même si les interruptions sont désactivées
* le code du debugger est isolé par le mode protégé du 80386. Celà empêche un programme 'errant' de modifier ou de détruire Soft-ICE; même si DOS se plante, Soft-ICE continue à fonctionner
* la possibilité de configurer Soft-ICE pour ne pas utiliser de mémoire conventionnelle si votre machine

possède plus de 640K
* une aide en ligne dynamique
* la possibilité d'être utilisé tout seul. Cette possibilité est interressante si vous débuggez des pilotes de périphérique, des gestionnaires d'interruption, ou des séquences de boot où les débuggers traditionnels n'ont pas accès ou si votre débugger a des problèmes de ré-entrance
* une possibilité de boot qui vous permet de débugger avec des systèmes d'opération non-DOS ou des programmes auto-bootables
* une installation simple, sans DIP à positionner, ni entrée/sortie utilisée, ni conflit d'addressage

NOTE:
Soft-ICE ne fonctionne pas avec les programmes en mode réel.
Il ne fonctionne pas avec les programmes utilisant les instructions
du mode protégé des 80286 et 80386

1.2 Utilisation de ce Manuel

Le manuel de Soft-ICE est divisé en quatre sections principales:
Apprentissage de Soft-ICE (Section I)
Commandes (Section II)
Caratéristiques (Section III)
Fonctions Avancées (Section IV)

Soft-ICE peut être utilisé pour beaucoup de problèmes de débuggage après lecture de le section I, "Apprentissage de Soft-ICE", et un peu d'expérimentation. L'aide en ligne de Soft-ICE peut être utilisée pour la description des commandes et leur syntaxe.

La section "Apprentissage de Soft-ICE" contient des instructions d'installation, une description de l'interface utilisateur et un didacticiel.
Le didacticiel est destiné à vous former rapidement.

La section "Commandes" décrit toutes les commandes de Soft-ICE.
Les descriptions de commandes sont organisées par groupes fonctionnels avec un index alphabétique de référence.

La section "Caractéristiques" recouvre les options avancées de chargement, le débugging source et symbolique et les possibilités EMM 4.0.

La section "Fonction Avancées" recouvre des sujets tels que l'utilisation de Soft-ICE avec des pilotes de périphérique et son utilisation avec des systèmes d'exploitation non-DOS.

Tout au long de ce manuel, particulièrement dans le didacticiel et dans la section consacrée aux commandes, des exemples sont donnés qui imposent de fournir à Soft-ICE des données. Quand les instructions vous demandent de "presser" une touche, par exemple la touche ' ', vous devez presser la touche marquée ' '. Quand les instructions vous demandent

d''entrer' une phrase telle que 'WIN', vous devez entrer les lettres citées, puis presser la touche ENTER.

1.3 Matériel Nécessaire

Soft-ICE fonctionne avec les IBM Séries II Modèle 70 et 80, Compaq 80386
et les ordinateurs à base de 80386SX, les compatibles AT et les cartes
co-processeur 80386. Soft-ICE ne fonctionne avec les co-processeurs 80386 XT que s'ils sont compatibles AT.

Soft-ICE fonctionne mieux avec de la mémoire étendue, mais fonctionne correctement avec de la mémoire conventionnelle.

Soft-ICE n'utilise ni DOS ni le BIOS pour ses affichages vidéo ni pour
ses entrées clavier. Malgrès ce, la vidéo doit être compatible avec l'un
des systèmes suivant: MDA, Hercules, CGA, EGA, VGA. Soft-ICE supporte aussi
les configurations à deux moniteurs, ce qui peut être très utile pour les
programmes faisant un usage intensif de la vidéo.

SECTION I - - Apprentissage de Soft-ICE

CHAPITRE 2

Début

2.1 Les Disquettes

Soft-ICE est livré sur une disquette 5 1/4" ou une disquette 3 1/2"
Quand vous lancez Soft-ICE, le nom de la personne possédant la license
de votre copie est affiché à l'écran pour contrer les copies pirates.
La disquette Soft-ICE n'est pas physiquement protégée contre la copie
pour votre confort. Pour notre confort, nous aprécierons l'attention
que vous porterez aux accords de license. Il est important de faire une
copie de sécurité de votre disquette avec pour seul usage le backup en
cas d'avarie de votre disquette originale.

Le catalogue de la disquette Soft-ICE montre les fichiers suivants:

```
S-ICE.EXE
S-ICE.DAT
LDR.EXE
MSYM.EXE
EMMSETUP.EXE
UPTIME.EXE
README.SI
SAMPLE.EXE
SAMPLE.ASM
SAMPLE.SYM
```

S-ICE.EXE est le programme Soft-ICE.
S-ICE.DAT est le fichier d'initialisation de Soft-ICE.
LDR.EXE est le loader de programmes et de fichiers symboles de Soft-ICE.
MSYM.EXE est le programme de création de fichiers symboles de Soft-ICE.
EMMSETUP.EXE est le programme qui vous permet de personnaliser la façon
dont votre système utilisera la mémoire paginée.
UPTIME.EXE règle l'heure DOS à celle de l'horloge CMOS.
README.SI est un fichier texte contenant des informations ne figurant
pas dans ce manuel.
SAMPLE.EXE est un court programme de démonstration utilisé par le didacticiel
SAMPLE.ASM est code source du programme de démonstration.
SAMPLE.SYM est le fichier symbole du programme de démonstration.

2.2 Chargement de Soft-ICE

Avant de lancer Soft-ICE, copiez tous les fichiers de la disquette de distribution sur votre disque dur.

Ces fichiers doivent être disponibles dans votre PATH.

S-ICE.EXE peut être chargé comme un pilote de périphérique dans votre CONFIG.SYS ou peut aussi être lancé comme un programme depuis la ligne
de commande DOS. Pour utiliser la plupart des possibilités de Soft-ICE,
il doit être lancé comme un pilote de périphérique dans votre CONFIG.SYS.

Note :
Si vous n'avez pas de mémoire étendue, Soft-ICE ne peut PAS être chargé
comme un pilote de périphérique. Il doit être lancé depuis le prompt du DOS.

2.2.1 Chargement sans Mémoire Etendue

Quand il n'y a pas de mémoire étendue présente, Soft-ICE se charge dans
l'emplacement mémoire le plus haut disponible. La mémoire utilisée par
Soft-ICE est ensuite cachée au DOS. Malgrès que la mémoire restant visible
sous DOS soit inférieure après le chargement de Soft-ICE, il est recommandé
de charger Soft-ICE avant tout TSR ou programme de contrôle.
Si vous n'avez pas de mémoire étendue, entrez simplement:

```
S-ICE
```

2.2.2 Chargement avec Mémoire Etendue

Le chargement de Soft-ICE avec de la mémoire étendue peut se faire d'une
des deux façons suivantes:

1. Installer S-ICE.EXE comme un pilote dans le fichier CONFIG.SYS.
   Cette méthode est nécessaire si vous souhaitez utiliser une des possibilités suivantes:
   * Partage de la mémoire avec des programmes utilisant la mémoire étendu par le biais d'appels à la ROM BIOS (VDISK.SYS, RAMDRIVE.SYS, HIMEM.SYS, programmes de cache, etc ..).
   * Utiliser le potentiel EMM 4.0 de Soft-ICE
   * Utiliser Soft-ICE pour le débuggage source ou symbolique.
   * Utiliser la trace arrière
   * Utiliser Soft-ICE avec d'autres produits Nu-Méga tels MagicCV,

   Quand il est chargé comme un pilote, Soft-ICE s'alloue une portion de mémoire étendue pour lui-même et pour ses fichiers annexes de telle sorte qu'il n'y ait pas de conflit mémoire.
   S-ICE .EXE doit être chargé depuis CONFIG.SYS avant tout autre driver allouant de la mémoire

étendue (ex: VDISK.SYS, RAMDRIVE.SYS). En général, Soft-ICE fonctionne mieux s'il est le premier pilote installé par CONFIG.SYS.
Pour les utilisateurs novices, il est conseillé de charger Soft-ICE comme premier pilote dans le CONFIG.SYS avec la commande suivante:

device = lecteur: \chemin\S-ICE.EXE /SYM 50

Lecteur et chemin spécifient le répertoire où se trouve Soft-ICE. Cette commande charge Soft-ICE lors de l'initialisation du système et sera très bonne pour le didacticiel.
Malgrès tout, Soft-ICE ne sera pas installé avec certaines de ses caractéristiques les plus avancées comme EMM 4.0. Vous pourrez reconfigurer Soft-ICE avec ces possibilités après avoir un peu expérimenté.
Si vous avez déjà l'expérience de Soft-ICE, ou si vous voulez installer ces possibilités immédiatement, reportez-vous s'il vous plait au chapitre 6 (Installation des Options de Soft-ICE).

Attention:
Lors de l'installation d'un pilote de périphérique pour la première fois sur votre système, il est prudent d'avoir à portée de main une disquette système. cette mesure prudente vous servira si, par malchance,
le réglage par défaut du pilote n'est pas compatible avec votre système.

Si vous n'êtes pas sûr de la façon d'éditer votre CONFIG.SYS, reportez-vous au guide de l'utilisateur de votre système ou au guide de l'utilisateur de votre éditeur de textes pour obtenir des instructions.
Après avoir modifié votre fichier CONFIG.SYS, vous devrez relancer votre système pour que les modifications prennent effet.

2. Lancez Soft-ICE depuis le prompt DOS en entrant S-ICE. Avant son chargement, Soft-Ice affiche un message de chargement et un prompt. Pour éviter ce prompt, incluez le mot EXTENDED dans le fichier S-ICE.DAT. Voir la section 6.4 pour plus d'informations sur le fichier S-ICE.DAT.
   En utilisant cette méthode, Soft-ICE est chargé automatiquement en haut de la mémoire étendue, qu'il y ait ou non quelque chose à cet endroit.
   Si vous êtes sûr de n'avoir aucun programme qui utilise la mémoire étendue, cette méthode est acceptable. Quand il est chargé avec cette méthode, Soft-ICE n'occupe AUNCUN octet en mémoire conventionnelle. La commande à utiliser est:

S-ICE

Notes:
Vous ne pourrez PAS activer toutes les possibilités de Soft-ICE en le chargeant depuis la ligne de commande. Si vous souhaitez utiliser Soft-ICE
comme débugger unique, il est recommandé de le charger depuis le fichier
CONFIG.SYS.

Si vous souhaitez charger Soft-ICE comme un pilote de périphérique

mais
que vous ne souhaitez pas que Soft-ICE reste en permanence résident, vous
devez utiliser l'option /UN au chargement. Voir la section 6.3.1 pour plus
d'informations.

2.2.3 Configuration de Soft-ICE pour une Installation Personnalisée

vous pouvez personnaliser Soft-ICE avec les options de chargement dans
le fichier CONFIG.SYS et le fichier de configuration S-ICE.DAT.
Les options du fichier CONFIG.SYS permettent de paramétrer comment la
mémoire étendue sera réservée par Soft-ICE. Le fichier d'initialisation
S-ICE.DAT vous permet de préciser des options de configuration, d'affecter
des commandes aux touches de fonction, et de définir une chaîne de
démarrage. Une chaîne de démarrage sert à exécuter une série de commandes
que vous utilisez chaque fois que vous installez Soft-ICE. Pour plus
d'informations sur la personnalisation de Soft-ICE, référez-vous au
chapitre 6.

2.3 Déchargement de Soft-ICE

Occasionnellement, vous pouvez être amené à vouloir décharger Soft-ICE.
Une des principales raison qui peut vous pousser à vouloir décharger
Soft-ICE, est de vouloir lancer un programme qui utilise les instructions
du mode protégé du 80286 ou du 80386. pour décharger Soft-ICE, entrez:

```
S-ICE /U
```

Cette commande replace la machine en mode d'adressage réel. Si Soft-ICE
avait été chargé depuis le fichier CONFIG.SYS, la mémoire réservée par
Soft-ICE, reste réservée et ne peut pas être utilisée par un autre
programme. Si Soft-ICE avait été chargé depuis la ligne de commande DOS,
le fait de le décharger libère la mémoire qui lui avait été allouée.

Attention:
Si de la mémoire étendue est en service au moment ou vous déchargez
Soft-ICE, ceci plantera votre système.

2.4 Recharger Soft-ICE

Soft-ICE peut être rechargé nimporte quand s'il avait été initialement
chargé depuis le fichier CONFIG.SYS. Si Soft-ICE avait été initialement
chargé depuis le fichier CONFIG.SYS, les options de configuration originales
sont conservées (EMM 4.0, symboles et source...).
Pour recherger Soft-ICE, entrez:

S-ICE

CHAPITRE 3

Débugger En 30 Minutes

3.1 Introduction

Toutes les interactions avec Soft-ICE prennent place dans une fenêtre qui peut être activée à tout moment. Toutes les commandes de Soft-ICE s'inscrivent dans une petite fenêtre, mais qui peut être agrandie jusqu'au plein écran si nécessaire. Vous utiliserez normalement la petite fenêtre lorsque vous emploirez Soft-ICE avec un autre débugger et la grande fenêtre lorsque vous l'emploirez tout seul.

La fenêtre est normalement ouverte plein écran si vous avez utilisé le fichier de configuration (S-ICE.DAT) inclus sur la disquette de distribution.

3.2 Activation de la fenêtre

Vous pouvez activer la fenêtre nimporte quand après avoir installé Soft-ICE. La séquence de touches prévue à cet effet est CTRL + D. Cette séquence peut naturellement être modifiée en utilisant la commande ALTKEY (voir section 5.8).

3.3 Sortie de la fenêtre

Le retour à l'affichage d'origine se fait grâce à l'utilisation de la commande X ou de la combinaison de touches utilisée pour activer Soft-ICE.
Tous les points d'arrêt que vous avez pu définir pendant que vous étiez
sous Soft-ICE demeurent actifs.

3.4 Changement de taille de la fenêtre

Vous pouvez modifier la largeur et la hauteur de la fenêtre de Soft-ICE.
Le changement de taille de la fenêtre est particulièrement utile en mode Soft-ICE seul lorsque vous affichez du code mémoire.

La hauteur de la fenêtre peut varier de 8 à 25 lignes. Pour changer cette
hauteur, utilisez les touches suivantes:

```
    ALT _ -- Agrandit la fenêtre
    ALT _ -- Diminue la fenêtre
```

Pour changer les dimensions de la fenêtre, utilisez la commande WIN (voir section 5.9). entrer WIN sans paramètre bascule entre ces deux modes:

```
   WIDE mode -- fenêtre plein écran
   NARROW mode -- 46 caractères
```

Certaines commandes (ex: D, E ,R ,U) tirent avantage des plus grandes dimensions en affichant plus d'informations quand la fenêtre est en mode WIDE.

3.5 Déplacement de la fenêtre

La fenêtre de Soft-ICE est déplaçable et peut être positionnée nimporte où
sur l'écran. Ceci est très utile quand la fenêtre est en mode NARROW.
Déplacez la fenêtre chaque fois que vous souhaitez voir des informations
sur l'écran qui se trouvent sous la fenêtre. Les touches suivantes seront
utilisées pour déplacer la fenêtre:

```
    CTRL _  -- remonte la fenêtre d'une case
    CTRL _  -- descend la fenêtre d'une case
    CTRL -> -- déplace la fenêtre d'une case vers la droite
    CTRL <- -- déplace la fenêtre d'une case vers la gauche
```

3.6 Touches d'édition de ligne

L'éditeur de ligne de Soft-ICE vous permet de rappeler et d'éditer les
commandes précédentes. Les fonctions de l'éditeur de ligne sont identiques
à celles du populaire éditeur de lignes CED.
Les séquences de touches suivantes vous aideront à éditer les commandes
dans le fenêtre de commande:

```
     ->     -- déplace le curseur vers la droite
     <-     -- déplace le curseur vers la gauche
    INS     -- bascule en mode insert
    DEL     -- efface le caractère courant
    HOME    -- déplace le curseur au début de la ligne
    END     -- déplace le curseur à la fin de la ligne
    _       -- affiche la commande précédente
    _       -- affiche la commande suivante
    SHIFT _ -- scrolle l'écran une ligne vers le haut
    SHIFT _ -- scrolle l'écran une ligne vers le bas
    PAGE UP -- scrolle l'écran une page vers le haut
    PAGE DN -- scrolle l'écran une page vers le bas
    BKSP    -- efface le caractère précédent
    ESC     -- annule la commande courante
```

Il y a des assignations de touches spéciales quand le curseur est dans la fenêtre de données ou dans la fenêtre de code. Elles sont décrites dans les sections concernant respectivement les commandes E et EC. Une assignation particulière à noter est SHIFT _ et SHIFT _ quand le curseur est dans la fenêtre de code. Ces touches sont réassignées pour avoir la même fonction que les touches _ et _ ont normalement. De cette facon, vous pouvez rappeler les précédentes commandes tandis que le curseur est dans la fenêtre de code.

3.7 Ligne d'Etat Interactive

Une ligne d'état au bas de la fenêtre fournit une aide interactive et la syntaxe des commandes.

3.8 Syntaxe des Commandes

Soft-ICE est un outil de débugging géré par ligne de commande. Pour interagir avec Soft-ICE, vous entrez des commandes, parfois modifiées par des paramètres.

Toutes les commandes sont des chaînes contenant de un à six caractères (en majuscules ou minuscules). Tous les paramètres sont des chaînes ASCII ou des expressions.

Les expressions sont normalement des nombres, mais peuvent aussi être des combinaisons de nombres et d'opérateurs (ex, + - / *). Tous les chiffres sont affichés en hexadécimal. Les paramètres octets sont affichés sur deux chiffres, les paramètres mots sur quatre et les paramètres double mot sont sous la forme de deux mots séparés par deux points (:). Voici quelques exemples de ces différends paramètres:

```
12 -- paramètre octet
10FF -- paramètre mot
E000:0100 -- paramètre double mot
```

Des registres peuvent être utilisés à la place des octets ou des mots dans une expression. Par exemple la commande 'U CS:IP-10' commence le désassemblage des instructions dix octets avant l'adresse du compteur d'instructions. Les registres suivants peuvent être utilisés dans une expression:

```
AL, AH, AX, BL, BH, BX, CL, CH, CX, DL, DH,
DX, DI, SI, BP, SP, IP, CS, DS, ES, SS, or FL
```

3.8.1 Spécification d'Addresses Mémoire

Beaucoup de commandes de Soft-ICE nécessitent des adresses mémoire comme paramètre. Une adresse mémoire est une valeur constituée de deux mots de 16 bits, séparés par deux points. Le premier mot est l'adresse de segment, le second mot est l'offset dans le segment.
Des symboles publics peuvent être utilisés à la place des adresses dans nimporte qyelle commande de Soft-ICE. Les symboles publics doivent avoir été chargés avec le programme chargeur (LDR.EXE). Voir le chapitre 7 (Symboles et Source) pour une description complète de l'utilisation des symboles publics.

L'évaluateur d'expressions de Soft-ICE reconnait certains caractères spéciaux en conjonction avec les adresses. Ces caractères spéciaux sont:

```
$        -- Courant CS:IP.
@address -- Indirection sur un Double Mot
.number  -- Numéro de Ligne du Source
```

Le caractère $ peut être utilisé à la place de CS:IP pour entrer l'adresse du pointeur d'instructions.

Le caractère @ vous permet de faire référence à un mot double pointé par une adresse. Vous disposez de plusieurs niveaux de @

Si le caratère . précède une adresse, celle-ci sera interprétée comme un numéro de ligne du fichier source courant, à la place de l'adresse actuelle.
Ceci n'est valable que si un fichier source a été chargé. L'adresse est interprétée en décimal dans ce cas là.

Exemples:

U.1234
Cette commande commence le désassemblage d'intructions a la ligne 1234 (décimal) du code source.

U $-10
Cette commande désassemble des instructions en commençant 10 octets avant le pointeur d'instructions.

G @SS:SP
Suppose que vous êtes sur la première instruction d'une routine d'interruption. L'entrée de cette commande crée un point d'arrêt temporaire à l'adresse de retour sur la pile et saute la routine d'interruption.

3.9 Touches de Fonction

Les touches de fonction peuvent être affectées à nimporte quelle

chaîne
de commande pouvant être entrée dans Soft-ICE. Les touches de fonction
peuvent être réaffectées depuis la ligne de commande ou préaffectées
dans le fichier de définitions S-ICE.DAT

Le fichier S-ICE.DAT livré avec votre disquette de distribution contient
les redéfinitions pour les 12 touches de fonction. Vous pouvez modifier celle
que vous voulez. Les redéfinitions fournies le sont à titre d'exemple,
mais elles ont été créées pour faciliter la vie aux utilisateurs de CodeView
de MicroSoft. Les affectations par défaut sont:

```
F1 -- Affiche l'aide générale (H;)
F2 -- Bascule la fenêtre registre ( ^WR;)
F3 -- Change le mode source courant ( ^SRC;)
F4 -- Restaure l'écran ( ^RS;)
F5 -- Retour à votre programme ( ^X;)
F6 -- Bascule le curseur entre la fenêtre de commandes
         et la fenêtre de code ( ^EC;)
F7 -- Va jusqu'au curseur ( ^HERE;)
F8 -- Pas à pas ( ^T;)
F9 -- Positionne un point d'arrêt sur la ligne courante ( ^BPX;)
F10 -- Trace ( ^P;)
F11 -- Aller à l'addresse de retour (modèle large)
          ( ^G@SS:SP;)
F12 -- Affiche le numéro de version de Soft-ICE ( ^VER;)
```

Un accent circonflexe (^) précédant une commande la rend invisible,
un point virgule suivant une commande représente un retour de chariot.
Vous pouvez afficher l'affectation courante des touches de fonction en
entrant la commande:

```
FKEY
```

Pour utiliser une touche de fonction, appuyez simplement sur la touche
au lieu d'entrer la commande. Pour reprogrammer les touches de fonction,
voir la section 5.8 pour une description de la commande FKEY, ou le
chapitre 6 pour une explication sur la pré-initialisation des touches
de fonction dans S-ICE.DAT.

3.10 Aide

La commande d'aide affiche un courte description, la syntaxe et un exemple
de chaque commande. Pour afficher l'aide, entrez:

```
? ou H -- affiche une brève description de toutes les
                  commandes et des opérateurs
? commande ou
H commande -- affiche une aide plus détaillée sur la
                  commande spécifiée, sa syntaxe, et
                  un exemple
? expression ou
H expression -- affiche la valeur de l'expression en
```

hexadécimal, décimal et ASCII

3.11 Didacticiel

Le didacticiel vous montre une bonne partie des possibilités de Soft-ICE
et vous donne la possibilité de l'essayer. Soft-ICE peut être utilisé conjointement avec un autre débugger ou bien tout seul. Le didacticiel
vous enseigne comment utiliser Soft-ICE avec un autre débugger, DEBUG,
et comment l'utiliser tout seul avec un source et des symboles chargés.
DEBUG peut être trouvé sur les disquettes système PCDOS ou MSDOS. Si vous ne possédez pas DEBUG, vous pouvez en utiliser un autre à la place
ou utiliser Soft-ICE tout seul.
Les utilisateurs souhaitant utiliser Soft-ICE pour un projet de "reverse engineering" ou pour débugger un pilote de périphérique DOS ou
un programme T&SR peuvent regarder ce didacticiel avec profit. Bien que
des exemples de ces types de programmes ne soient pas explicitement donnés, vous obtiendrez ainsi une vue d'ensemble du travail avec Soft-ICE. Il est recommandé d'expérimenter avec Soft-ICE et votre environnement propre avant d'entreprendre un projet réel.

Un court programme en langage assembleur est utilisé pour démontrer les points d'arrêt style hard. Le programme d'exemple est intentionnellement
court et parfait pour ceux qui ne seraient pas très familiers avec le langage assembleur. Ce didacticiel est destiné à vous donner un aperçu
des possibilités de Soft-ICE. N'hésitez pas à expérimenter après avoir
fait le tour du didacticiel.

Soft_ICE est très adaptable. Il permet d'être chargé de la meilleure façon possible pour votre système. Regardez les procédures d'installation
dans la section 2.2 avant de continuer avec le didacticiel.

Si vous n'avez pas de mémoire étendue sur votre système, vous devez charger Soft-ICE depuis la ligne de commande. Quand vous le chargez ainsi,
vous ne pouvez charger ni symboles, ni fichiers source. Dans ce cas, vous
devrez improviser dans la dernière section du didacticiel
où il est expliqué comment utiliser Soft-ICE tout seul.

Soft-ICE peut être chargé depuis le prompt du DOS ou comme pilote de périphérique dans votre fichier CONFIG.SYS. Pour les besoins de ce didacticiel, vous devez installer Soft-ICE dans votre CONFIG.SYS avec au moins 50K de mémoire étendue réservée pour les symboles et les fichiers sources. Soft-ICE doit être le premier pilote installé dans
le CONFIG.SYS. La ligne d'installation doit être:

```
DEVICE = drive:\path\S -ICE.EXE /SYM 50
```

Le paramètre /SYM 50 demande à Soft-ICE de réserver 50 kilooctets

de mémoire étendue pour le chargement des symboles et des fichiers source. C'est généralement insuffisant pour la plupart des problèmes courants, mais fonctionne avec notre exemple.

Vous devez redémarrer votre machine après ajout de cette ligne dans votre CONFIG.SYS.

Quand vous redémarrez votre système, Soft-ICE affiche un avertissement
de Copyright, un numéro d'enregistrement, le nom de la personne à qui appartient cette copie de Soft-ICE et la mémoire étendue réservée pour chacune des composantes de Soft-ICE. Avec un système disposant de
384K de mémoire étendue, l'écran initial doit ressembler à ceci:

Soft-ICE Your Name Your Company Name Registration # SInnnnnn
Copr. (C) Nu-Mega Technologies 1987-1989 All Rights Reserved Soft-ICE Version 2.00 Soft-ICE is loaded from 00132000H up to 00160000H. 50K of
symbol space reserved. 10K of back trace space reserved. 200 K of extended
memory available.
Le message "Soft-ICE is loaded ..." vous renseigne sur la zone exacte de
mémoire occupée par Soft-ICE et ses composantes. Si vous possédez un Compaq
ou une machine compatible avec Compaq, et si vous avez inclus le mot COMPAQ
dans votre fichier S-ICE.DAT, vous verrez aussi un message disant: "Using high memory from XXXXXXXX to 00FE0000H".

La ligne suivante vous indique combien de place a été réservée pour les
symboles. Cette place est utilisée pour les symboles et les fichiers source.

La ligne suivante vous indique combien de place a été réservée pour l'historique de trace arrière. La valeur par défaut est 10K. Cette place
est utilisée par la commande SNAP et la commande BPR avec les options T ou TW.

La dernière ligne vous indique combien de mémoire reste disponible en tant que mémoire étendue. Cette mémoire peut être utilisée par les autres programmes tels HIMEM, SMARTDRIVE, VDISK, etc ...

Positionnez-vous dans le répertoire ou vous avez copié les fichiers de
votre disquette de distribution. Souvenez-vous que ce répertoire doit figurer dans votre PATH.

Avant de nous lancer dans le débugging, faisons aparaître la fenêtre Soft-ICE et familiarisons-nous avec.

Effacons l'écran en tapant:

CLS

Affichons la fenêtre en pressant:

CTRL D

La fenêtre de Soft-ICE est maintenant à l'écran. Si votre fichier S-ICE.DAT a été trouvé, la fenêtre occupe l'écran tout entier.
Il est divisé en quatre sections. Du haut vers le bas, ces sections sont: la fenêtre des registres, la fenêtre des données,
la fenêtre de code et la fenêtre de commande.

Si S-ICE.DAT n'a pas été trouvé, vous avez une petite fenêtre au centre
de votre écran. Ceci peut aussi arriver si d'autres composants nécessaires
au didacticiel n'ont pas été chargés.

Si la petite fenêtre est visible, vous devez:

1. Quitter Soft-ICE en entrant un X.
2. Décharger Soft-ICE en tapant S-ICE /U.
3. Copier S-ICE.DAT depuis les disquettes de ditribution
   vers un répertoire accessible dans votre PATH.
4. Relancer la démo.

Nous voulons maintenant basculer vers la petite fenêtre.
La petite fenêtre est très utile lorsqu'on utilise Soft-ICE en
tant qu'accessoire d'un autre débugger.

```
    Entrez:
      WIN
```

Ceci fait apparaître la petite fenêtre de commande au centre de l'écran.
Certaines commandes de Soft-ICE sont visibles sur cet écran. Ce sont les
traces de la chaîne d'initialisation de S-ICE.DAT qui avait placé Soft-ICE
en mode plein écran au démarrage.

Vous pouvez remarquer un prompt (:) et la ligne d'état au bas de la fenêtre.

La fenêtre Soft-ICE peut être déplacée sur l'écran et sa taille ajustée.

```
   Déplacez la fenêtre sur l'écran en pressant:
        CTRL _  -- déplace la fenêtre d'une case vers le haut
        CTRL _  -- déplace la fenêtre d'une case vers le bas
        CTRL <- -- déplace la fenêtre d'une case vers la gauche
        CTRL -> -- déplace la fenêtre d'une case vers la droite
   Basculez la fenêtre en plein écran en tapant:
      WIN
```

Vous remarquez que l'écran d'origine est restauré.

```
  Revenez à la petite fenêtre en tapant WIN à nouveau.
  Agrandissez ou réduisez la fenêtre en tapant:

      ALT _ -- agrandit la fenêtre
      ALT _ -- réduit la fenêtre
```

Essayez maintenant ce qui vient naturellement à l'esprit lorsqu'on
se trouve face à un nouveau programme et qu'on ne sait pas quoi faire --
demandez de l'aide.

Affichez l'aide en tapant:

```
        ?
```

Remarquez que l'écran marque une pause et attend une touche avant
d'afficher la suite des informations à l'écran. Regardez la ligne d'état
au bas de la fenêtre. Elle affiche les instructions suivantes:

```
           "Any Key To Continue, ESC to Cancel ".
```

Pressez maintenant une touche pour afficher la suite de l'aide. Continuez
à presser une touche jusqu'au retour du prompt (:).

Revenez en arrière dans l'aide en pressant
_ et _

Les informations affichées précédement dans la fenêtre de commande peuvent être balayées avec les touches shift haut, shift bas, page haut
et page bas. Essayez plusieurs de ces touches pour balayer l'aide.
L'aide de Soft-ICE vous donne un apperçu de chaque commande. Si vous entrez un point d'interrogation (?) suivi d'un nom de commande, vous verrez la syntaxe de cette commande, un brève description et un exemple.

Faites donc des essais en entrant ces commandes:
? commande
Par exemple:
? ALTKEY

Faites attention à ce qui s'affiche sur la ligne d'état en bas de l'écran
si vous avez des doutes.

La commande ? vous permet aussi d'évaluer des expressions hexadécimales.

Par exemple, entrez:
? 10*2+42

Le résultat est affiché sous forme hexadécimale, décimale, puis en ASCII:
0062 00098 "b"

Vous activez la fenêtre par la séquence CTRL D. C'est parfait dans la majorité des cas, mais certains peuvent préférer une autre séquence.

Nous allons donc entrer une commande pour changer la séquence de touches
nécessaire pour activer la fenêtre. Procédez par étapes, vous serez aidés
par la ligne d'état au bas de la fenêtre.
Entrez la lettre 'A'. La ligne d'état affiche une liste des commandes commançant par 'A'. Continuez à taper le mot 'ALTKEY'. La ligne d'état affiche maintenant une brève description de la commande /ALTKEY. Pressez la barre d'espace. La ligne d'état montre maintenant la syntaxe requise pour la commande /ALTKEY. Entrez les lettres 'ALT D' et pressez ENTER pour valider la commande:
ALTKEY ALTD

Vous venez de changer la séquence de démarrage en ALT D.
A partir de maintenant il vous faudra presser ALT D pour activer la fenêtre. Ceci sera considéré comme vrai pour le reste du didacticiel.
Testons maintenant la commande.

Pour quitter la fenêtre, pressez:
ALT D

La fenêtre Soft-ICE a disparu.
Pour réactiver Soft-ICE, relachez ALT
et tapez:

ALT D

La fenêtre est réouverte.

Pour voir certaines des commandes précédentes, pressez _ plusieurs fois.

Remarquez que Soft-ICE se souvient des dernières commandes entrées.
Essayez d'en éditer une juste pour voir.
Les touches d'édition sont:

```
INS  -- Bascule le mode Insertion ON ou OFF
DEL  -- Efface un caractère
HOME -- Déplace le curseur au début de la ligne
END  -- Déplace le curseur à la fin de la ligne
->   -- Déplace le curseur d'une case vers la droite
<-   -- Déplace le curseur d'une case vers la gauche
```

Quand on est en mode insertion, notez le curseur en forme de pavé.

Maintenant que vous voilà un peu plus familiarisé avec l'environnement,
essayons quelques nouvelles commandes.

Effacez la commande que vous éditiez en pressant la touche HOME, puis le touche DEL jusqu'à effacement de la commande.

Entrez:

WR

La commande WR rend la fenêtre de registre visible. La fenêtre de registre affiche le contenu des registres du 8086. Remarquez que la valeur des registres reflète l'endroit où le code était exécuté quand vous avez activé Soft-ICE.

Augmentez la taille verticale de la fenêtre Soft-ICE avec les touches ALT et _ Jusqu'à ce qu'elle occupe tout l'écran. Notez les valeurs de registres CS et IP dans la fenêtre registre, puis tapez:

MAP

La commande MAP affiche une carte de la mémoire système. La zone dans laquelle se trouve le pointeur d'instructions (CS:IP) est en surbrillance. Si votre carte mémoire est complexe, il vous faudra plusieurs appuis touche pour retrouver le prompt.

Essayez maintenant la séquence suivante plusieurs fois en notant la valeur du registre (CS:IP).

ALT D
Relachez ALT et D
ALT D

Chaque fois que la fenêtre s'affiche, vous pouvez remarquer que la valeur
des registres CS et IP a changé. Quand CS et IP changent, vous pouvez entrer la commande MAP à nouveau pour voir si le pointeur d'instructions
pointe sur une zone différente.

Ce petit exercice démontre que Soft-ICE est un débugger que l'on peut activer où que soit le pointeur d'instructions lorsqu'on presse la séquence de touches qui permet de lancer Soft-ICE. Le pointeur d'instructions change continuellement car il se passe une foule de

choses dans votre machine même si vous êtes sous le prompt du DOS, telles que gestion des interruptions, surveillance des pilotes de périphériques, etc ...

Pressez la touche de fonction F12.

La touche de fonction F12 est assignée par défaut à la commande VER. Cette commande affiche le message de copyright de Soft-ICE et son numéro de version.
Nous allons maintenant assigner à la touche F12 la commande RS.

Entrez:
RS

Ceci affiche temporairement l'écran du programme en cours d'exécution sans la fenêtre Soft-ICE.
Pressez la barre d'espace pour revenir à la fenêtre Soft-ICE.

Entrez:
FKEY F12 RS;

Ceci assigne la commande RS à la touche F12. Le point virgule repésente
la touche ENTREE.

Pressez la touche F12.

Recommencez plusieurs fois pour basculer entre la fenêtre de Soft-ICE et l'écran du programme. Soyez sûr que la fenêtre Soft-ICE est affichée,
au besoin en appuyant sur F12. Remarquez la commande RS affichée plusieurs
fois dans la fenêtre. Elle est affichée autant de fois que la touche F12
a été pressée.

Nettoyez la fenêtre Soft-ICE en entrant:
CLS

Entrez:
FKEY F12 ^RS;

Le symbole ^ est ob tenu en pressant shift + 6. Ceci affecte la commande RS à la touche de fonction F12 mais rend la commande invisible.

Pressez la touche F12 quelques fois. Remarquez que
la commande RS ne s'affiche plus dans la fenêtre.

Vous pouvez aussi affecter une suite de commandes Soft-ICE à une touche de fonction. N'oubliez pas de mettre un retour de chariot entre chaque commande.
Maintenant, préparez-vous à utiliser Soft-ICE comme un assistant de l'utilitaire DEBUG de MSDOS.

Diminuez la hauteur de la fenêtre de 6 lignes environ
à laide des touches ALT _.
Entrez:
ACTION INT3

Cette commande indique à Soft-ICE de générer une INT3 chaque fois que les conditions d'un point d'arrêt sont satisfaites. C'est ainsi que Soft-ICE communique avec DEBUG. Le réglage par défaut est HERE.
ACTION HERE rend le contrôle directement à Soft-ICE. Utilisez ACTION

HERE quand vous utilisez Soft-ICE tout seul.

Pour ceux qui n'utilisent pas DEBUG avec ce didacticiel, le moment est
venu d'improviser. CODEVIEW fonctionne avec ACTION réglé sur NMI. Beaucoup d'autres débuggers fonctionnent avec ACTION réglé sur INT3. Si ce n'est pas le cas de votre débugger et que vous souhaitiez de l'aide, reportez-vous à la description complète de la commande ACTION
(voir section 5.4).

Pour faire à nouveau disparaître la fenêtre Soft-ICE, entrez:
X

Ceci est une autre méthode pour sortir de SOft-ICE. Elle est spécialement pratique avec les touches de fonction.

Maintenant que vous êtes familiarisés avec les bases de l'utilisation de Soft-ICE, nous allons approfondir quelques points en débuggant un programme simple (SAMPLE.ASM).

SAMPLE.ASM est un programme simple écrit en langage assembleur par un programmeur nommé Jed. Ce programme lit une touche au clavier et affiche un message disant si la touche pressée était ou non un espace.

Pour lancer le programme SAMPLE, entrez:
SAMPLE

Maintenant, pressez la barre d'espace. Pressez d'autres touches. Apparement, le programme de Jed a un problème! Jed a passé plusieurs heures à réviser son code source et il est sûr qu'il n'y a pas de faille dans sa logique. Par contre, Jed a récupéré certaines routines d'un de ses amis, Jake (get_key, is_space?). Jed est quelque peu méfiant
à leur propos mais ne trouve pas le bug.

Le code source du programme de Jed est celui-ci:

```
   Page 55,80
   Title Sample program for Soft-ICE tutorial
 DATA Segment Public 'Data'
 pad       db 12H dup(O)
 char db 0
 answer db 0
 space_msg db 'Le Caractère est un SPACE',0DH,0AH,'$'
 no_space_msg db 'Le Caractère N'EST PAS un'
             db 'SPACE',0DH,0AH,'$'
 DATA Ends
 STACK Segment Stack 'Stack'
   Dw 128 Dup (?)     ;Program stack STACK Ends
 CODE Segment Public 'Code'
 Assume CS:CODE,DS:DATA,ES:Nothing,SS:STACK

start:


; Set up segments
   mov ax,DATA
   mov es,ax
   mov ds,ax

; Main Program Loop
main,loop:
```

```
    call get_key
    call is_space?
    cmp answer,0
    je no,space
; Si c'est un Espace, on affiche le message Espace

   mov ah,9
   mov dx,offset space_msg
   int 21H
   jmp main_loop
; Si ce n'est pas un Espace,on affiche le message pas Espace

no_space:
   mov ah,9
   mov dx,offset no_space_msg
   int 21H
   jmp main_loop

;-----------------------------------------------------------;
; JAKE'S ROUTINES
;-----------------------------------------------------------;

; Get Key Routine (Une des routines de Jake)
get_key proc
   mov ah,8
   int 21H
   mov char,al
   ret get_key endp

; Check if character is a space (Une des routines de Jake)

is_space? proc
   cmp char,20H
   jne not_space
   mov answer, 1
   ret not_space:
   mov cs:answer,0
   ret is_space? endp

CODE Ends
   Endstart
```

Jed utilise DEBUG, mais n'a pas été capable de résoudre le problème.
Sur les conseils d'un ami, Jethro, Jed a acheté Soft-ICE. Il était assez
réticent à l'utiliser car il avait déjà essayé un debugger Hard mais n'avait
jamais réussi à le faire fonctionner corectement. Il était décidé à utiliser
Soft-ICE, car il lui permettait de continuer à se servir de DEBUG, le seul
débugger qu'il maîtrise vraiment.

```
   Pressez CTRL C pour sortir du programme.
    Entrez la commande suivante:
     DEBUG lecteur:\chemin\SAMPLE.EXE
       U
       R
```

Durant les heures passées à essayer de trouver ce bug, Jed a acquit la
conviction que quelque chose écrase son code de façon curieuse.
Avec Soft-ICE, Jed décide de positionner un point d'arrêt domaine mémoire

autour de son segment de code.

Pressez:
ALT D

La fenêtre de Soft-ICE apparaît. Déplacez la fenêtre (à l'aide des touches CTRL et des flèches) jusqu'à faire apparaître l'affichage des registres sous DEBUG. Il est temps de positionner votre premier Point d'Arrêt.

Entrez:
BPR code-seg:0 code-seg:25 W

Code-seg est la valuer du registre CS comme affiché par la commande R de DEBUG.

La commande BPR positionne un point d'arrêt sur un domaine mémoire. La longueur du segment de code de Jed est de 25H octets, donc la zone mémoire spécifiée va du début de ce segment de code à la fin. Le W sert
à indiquer à Soft-ICE de réagir sur une écriture. Nous souhaitons surveiller toute écriture intempestive sur le code de Jed.

Entrez:
BL

La commande BL affiche tous les points d'arrêt. L'affichage de BL doit ressembler à ceci:

```
0) BPR code-seg:0000 code-seg:0025 W C = 01
```

Le 0 est un identifieur pour ce point d'arrêt. Le domaine et W sont affichés tels qu'ils ont été entrés, et le count (si rien n'a été précisé), est par défaut à 1.

Voici le moment de vérité.
Pressez ALT D.

La fenêtre disparaît à nouveau.
Pour lancer SAMPLE depuis DEBUB, entrez:
G
Pressez la barre d'espace. OK. Maintenant pressez une autre touche.

Notre point d'arrêt nous a renvoyés dans DEBUG. Les registres et une seule instruction désassemblée sont affichés.

Entrez:
U cs:addresse

Addresse est la valeur du registre IP moins 10H. Comme DEBUG est un peu primitif, la valeur de IP moins 10H doit être calculée à la main. Le compteur d'instructions pointe sur l'instruction qui est juste après celle ayant causé le point d'arrêt. En revenant en arrière de 10H instructions, DEBUG peut se resynchroniser.

L'instruction à l'offset 3BH est:

```
CS:
MOV BYTE PTR [13],0
```

Jed s'exclame,"Il est là! Je savais que le problème était dans les routines de Jake! Cette instruction d'écriture dans le segment de code
met un octet à 0 dans mon code. Qui peut se douter du résultat !"

Entrez:
U 0

L'emplacement 13H est en fait l'offset d'une instruction de saut conditionnel. L'offset relatif du saut conditionnel est ainsi mis à 0.

Maintenant, nous allons voir comment ce problème peut être résolu en utilisant Soft-ICE tout seul. Nous devons tout d'abord quitter DEBUG.

Avant de quitter le débugger, il est toujours souhaitable de désarmer tous les points d'arrêt à moins que ACTION soit positionné sur HERE. Si vous ne le faites pas et qu'un point d'arrêt se produise, ACTION va tenter de réactiver un débugger qui n'est plus en mémoire; le résultat est imprévisible. Nous avons positionné ACTION sur INT3, nous allons donc désarmer le point d'arrêt.

Pour afficher la fenêtre, pressez:
ALT D

Listez les points d'arrêt en entrant:
BL

Remarquez que la ligne contenant la description du point d'arrêt est en surbrillance. Le point d'arrêt en surbrillance est le dernier à s'être produit.

Notez que le point d'arrêt porte le numéro 0.
Pour désarmer le point d'arrêt numéro 0, entrez:
BD 0
Listez les points d'arrêt à nouveau en entrant:
BL

L'astérisque (*) après le numéro du point d'arrêt montre que le point d'arrêt est désactivé.

Pour effacer un point d'arrêt, entrez:
BC 0
Entrez BL à nouveau.

Remarquez qu'il n'y a plus de point d'arrêt affiché.

quittez Soft-ICE, puis quittez DEBUG, en entrant:
X
Q

La partie suivante du didacticiel montre comment Soft-ICE peut être utilisé tout seul pour résoudre le même genre de problème. Soft-ICE sera utilisé comme débugger source.

Pour préparer Soft-ICE à débugger au niveau source, il doit avoir été installé dans votre fichier CONFIG.SYS et de la mémoire étendue doit avoir été allouée pour les symboles le fichier source. Soft-ICE ne peut être utilisé comme débugger source que si vous avez de la mémoire étendue sur votre système.

Si vous n'avez pas de mémoire étendue, vous pouvez tout de même lire le reste du didacticiel pour voir les possibilités de Soft-ICE avec

de la mémoire étendue. Si vous n'avez pas chargé S-ICE.EXE dans votre CONFIG.SYS avec de la mémoire réservée pour les symboles, faites-le maintenant.

Pour débugger le programme d'exemple avec Soft-ICE seul, vous devez utiliser le programme chargeur (LDR.EXE). Pour charger le programme exemple (SAMPLE.EXE), le fichier symbole (SAMPLE.SYM) et le fichier source (SAMPLE.ASM), entrez:

LDR SAMPLE

Vous êtes maintenant sous Soft-ICE avec SAMPLE.EXE chargé en mémoire. Remarquez que Soft-ICE occupe tout l'écran. Soft-ICE bascule en mode WIDE quand un programme est chargé. Le source (SAMPLE.ASM) est visible
dans la fenêtre code. De plus, les fenêtres données et registres sont visibles.

Exécutez une instruction en pressant F10.

Remarquez que la barre en vidéo inverse se déplace sur la prochaine instruction à exécuter après un saut.

Pressez F6.

Ceci place le curseur dans la fenêtre code.
Faites maintenant des essais avec les
touches _, _, pageUp, et pageDn
pour déplacer le curseur et vous déplacer
au long du source.
Déplacez le curseur sur la ligne 42.

Pressez F9.
Vous venez de positionner un point d'arrêt sur la ligne 42. La ligne est en surbrillance vous montrant ainsi qu'un point d'arrêt y a été placé.

Entrez:
BL

Cela vous montre le point d'arrêt qui vient juste d'être posé.

Maintenant, pressez ALT D.

Ceci sort de Soft-ICE et exécute le programme SAMPLE jusqu'au point d'arrêt de la ligne 42. Soft-ICE réapparait immédiatement avec la barre en vidéo inverse sur la ligne 42.

Pressez F6 à nouveau.

Ceci vous remet le curseur dans la fenêtre de commandes. Entrez maintenant:
BC *

Ceci efface tous les points d'arrêt (il n'y en avait qu'un).
Quittez maintenant Soft-ICE par ALT D.

Vous êtes de retour dans le programme exemple. Tapez quelques touches juste pour vous assurer qu'il ne fonctionne pas.
Retournez dans Soft-ICE par ALT D.

Comme le bug s'est déjà produit, vous devez relancer le programme. Entrez:
EXIT RD

Cette commande force le programme exemple à se terminer. Le R indique à Soft-ICE de restaurer les vecteurs d'interruption dans l'état où ils étaient quand le programme exemple a été chargé avec LDR. Le D indique à Soft-ICE d'effacer tous les points d'arrêt pouvant
encore se trouver en mémoire. Le R et le D ne sont pas nécessaires dans ce cas, mais c'est une bonne habitude à prendre avant de quitter un programme chargé par LDR.EXE.

Vous êtes de nouveau sous DOS. Rechargez le programme en entrant:

LDR SAMPLE.EXE

Remarquez que le suffixe .EXE a été précisé cette fois-ci. Quand le suffixe est spécifié, Soft-ICE n'essaie pas de charger les symboles ou le source. Dans notre cas, les symboles et le source sont toujours en mémoire.

Entrez:
SYM

Ceci affiche les symboles publics du programme exemple.
Pressez ESC pour revenir au prompt.

Nous voulons maintenant positionner un point d'arrêt domaine mémoire similaire à celui utilisé lors de la session avec DEBUG. Cette fois-ci,
nous utiliserons les symboles pour positionner le point d'arrêt. Entrez:

BPR START .82 W

Ceci positionne un point d'arrêt domaine mémoire dans votre segment de code, du symbole START à la ligne 82 du fichier source.

Entrez:
BL

Vous pouvez vérifier que le point d'arrêt a été positionné correctement.

Pressez ALT D.
Pressez une touche autre que Escape.

Vous voilà de retour dans Soft-ICE. Reamrquez que l'instruction courante
(la ligne avec la barre en vidéo inverse) est l'instruction qui suit celle
ayant provoqué le point d'arrêt.

Pour voir le code actuel, pressez F3.

Ceci place Soft-ICE en mode mixte. Notez que la barre en vidéo inverse
recouvre deux lignes. Il y a la ligne de code actuelle et la ligne de source correspondante.

Pressez la touche F3 à nouveau.

Vous êtes de nouveau en mode code. Les lignes de source ne sont plus visibles. L'instruction au-dessus de la barre en vidéo inverse est

celle
ayant causé le point d'arrêt.

Pressez la touche F3 à nouveau pour revenir en mode source.

Vous pouvez maintenant corriger le bug dans le programme d'exemple.
Sortez du programme d'exemple et revenez sous DOS
en entrant:
EXIT RD

Rechargez le programme exemple en entrant:
LDR SAMPLE. EXE

Positionnez la fenêtre de code en mode code en pressant
2 fois la touche F3.
Désassemblez la routine défaillante en entrant:
U not_space

Vous allez maintenant utiliser l'aasembleur interactif de Soft-ICE
pour corriger le problème.

Entrez:
A not_space

Soft-ICE vous répond en affichant l'adresse.
Entrez:
NOP
Pressez ENTER pour sortir de l'assembleur.

Remarquez, dans la fenêtre de code, l'instruction NOP à la place de
CS:
à l'offset 003BH.

Pressez la touche F3 pour revenir en mode source,
(le code source n'est bien sur pas modifié).
Pressez ALT D pour lancer le programme corrigé.

Entrez:
des espaces et quelques autre touches

Ca marche! Vous avez corrigé le bug!
Pour sortir du programme de Jed et revenir sous DOS, pressez:
CTRL C

Nous allons maintenant découvrir une autre caractéristique de Soft-
ICE.
Entrez:
LDR SAMPLE.EXE

Ceci recharge une fois encore le programme exemple.

Entrez:
RIP HANG_EXAMPLE

Les deux premières instructions affichées sont:
CLI
JMP $

Remarquez que l'instruction de saut pointe sur elle-même. Cette
boucle
infinie plante normalement le système.

Entrez:
BREAK ON

Vous venez d'activer le mode BREAK. Le mode BREAK fait tourner le système
un peu mois vite mais permet à Soft-ICE de reprendre la main même si le
système est bloqué.

Quittez Soft-ICE en pressant ALT D.
Votre système est maintenant bloqué. Pour les incrédules, tapez:
CTRL ALT DEL

Il ne se passe rien. C'est définitivement bloqué.
Maintenant pressez ALT D.

La fenêtre de Soft-ICE est de retour!
Pour quitter la boucle sans fin, entrez:
EXIT RD

Vous êtes maintenant à nouveau sous DOS. Faites quelques DIR pour bien
vous rendre compte de la dégradation des performances. Certaines
personnes ne trouvent pas génant de rester par défaut en mode BREAK.

Remettez le mode BREAK OFF en entrant:
ALT D
BREAK OFF
ALT D

Refaites quelques directories pour comparer la vitesse.
Voilà! C'est terminé! Il est temps d'expérimenter et de débugger par
vous-même. Feuilletez le reste du manuel et référez-vous à certaines
sections si nécessaire

SECTION II -- Commandes

La Section II contient la liste des syntaxes de chaque commande de
Soft-ICE, des explications et des exemples pour chaque commande.
Tous les nombres sont en hexadécimal; tout nombre peut être une
expression utilisant +,-,*,/, ou des registres. Les commandes
peuvent être entrées en majuscules ou en minuscules.
Les parties en italique dans l'exemple de syntaxe doivent être
remplacées par leur valeur actuelle.

Les conventions de notation suivantes sont utilisées tout au long de
cette section:

[ ] -- Les Crochets entourent un paramètre optionnel.

< > -- Entourent une liste d'objets ou de choix.

x | y -- Les barres Verticales séparent plusieurs alternatives. utilisez soit l'objet x soit l'objet y.

count -- Count est une valeur sur un octet qui précise le nombre de fois que les conditions d'un point d'arrêt doivent être réunies pour que celui-ci se produise. Si count n'est pas précisé, la valeur par défaut est 1. Chaque fois que la fenêtre Soft-ICE est activée, les compteurs count sont remis à leurs valeurs de départ.

verb -- Verb est une valeur qui précise quel type d'accès un point d'arrêt doit surveiller. Il peut être positionné à 'R' pour lecture, à 'W' pour écriture, à 'RW' pour lecture/écriture, ou à X pour exécute.

address -- Address est une valeur constituée de deux mots de 16 bits séparés par deux points (:). Le premier

mot est l'adresse de segment, le second est
l'adresse d'offset.

Les adresses peuvent contenir des registres et des symboles.
Les adresses peuvent aussi contenir les caractères spéciaux "$", "." et "@". Voir la section 3.8 (Syntaxe des commandes) pour une description de ces caractères spéciaux.

break-number -- Break-number est un nombre d'identification qui identifie le point d'arrêt. (Ex: pour l'éditer, l'effacer ou le désactiver). Ce nombre peut être un nombre hexadécimal compris entre 0 et F

list -- List est une série de numéros de points d'arrêt séparés par des espaces ou des virgules.

mask -- Mask est un masque de bits représenté par une combinaison de 1, de 0, et de X. Les X représentant les bits dont on ne tient pas compte.

Exemple:

```
BPIO 21 W EQ M 1XXX XXXX
```

Cette commande exécute un point d'arrêt si une écriture a lieu sur le port 21H avec le bit de poids fort positionné.

GT, LT -- GT et LT sont des qualifieurs des commande utilisés pour les comparaisons de valeurs.

CHAPITRE 4

Utilisation des Commandes de Point d'arrêt

4.1 Introduction

Soft-ICE a des possibilités de point d'arrêt qui ne sont normalement disponibles qu'avec des débuggers hard.

La puissance et la flexibilité du 80386 permettent des possibilités de point d'arrêt avancées sans équipement matériel supplémentaire.

Les points d'arrêt peuvent être positionnés sur les lectures ou les écritures en mémoire, les lectures ou écritures sur un domaine mémoire, l'exécution d'une instruction et les accès aux ports I/O. Soft-ICE attribue un chiffre hexadécimal (0-F) à chaque point d'arrêt. Ce nombre est utilisé pour identifier les points d'arrêt quand vous les positionnez, effacez, désactivez, activez ou éditez.

Tous les points d'arrêt de Soft-ICE sont permanents. Ils ne disparaissent pas automatiquement après avoir été employés; vous devez intentionnellement les effacer ou les dévalider en utilisant les commandes BC ou BD. Soft-ICE peut gérer 16 points d'arrêt en même temps.

Vous pouvez avoir jusqu'à dix points d'arrêt d'un même type à l'exception des points d'arrêt mémoire (BPM), desquels vous ne pouvez avoir que quatre, à cause des limitations du processeur 80386.

Les points d'arrêt peuvent être établis avec un paramètre 'count'. Le paramètre 'count' indique à Soft-ICE combien de fois le point d'arrêt doit être ignoré avant que le déclenchement survienne.

4.2 Positionner les points d'arrêt

Commandes:

```
BPM, BPMB, BPMW, BPMD -- positionne un point d'arrêt sur
                         un accès mémoire ou une exécution.

BPR -- Positionne un point d'arrêt sur un domaine mémoire.
BPIO -- Positionne un point d'arrêt sur un accès à un port I/O.
BPINT -- Positionne un point d'arrêt sur une interruption.
BPX -- Positionne/Efface un point d'arrêt sur une exécution.
CSIP --  Définit un domaine pour CS:IP.
BPAND -- Attend l'occurence de plusieurs points d'arrêt.
```

BPM, BPMB, BPMW, BPMD

```
BPM, BPMB, BPMW, BPMD -- positionne un point d'arrêt sur un accès
                         mémoire ou une exécution.

  Syntaxe:
    BPM [taille] adresse [verbe] [qualifieur] [C=count]
    taille -- B, W, D
                    B -- Octet
                    W -- Mot
                    D -- Mot Double
                    La taille est le domaine actuellement
                    couvert par ce point d'arrêt. Par
                    exemple, si un mot double est employé,
                    et le troisième octet du double est
                    modifié, alors un point d'arrêt
                    surviendra.
                    La taille est aussi importante si
                    l'option qualifier est utilisée
                    (voir ci-dessous).
    verbe -- R, W, RW, ou X
    qualifieur -- EQ, NE, GT, LT, M
                    EQ -- Egal
                    NE -- Différent
                    GT -- Plus grand que
                    LT -- Plus petit que
                    M -- Masque
                    Ces qualifieurs sont seulement
                    applicables aux points d'arrêt lecture
                    et écriture.
    valeur -- Un octet, un mot, ou un double mot,
                      en fonction de la taille spécifiée.
```

Commentaires:

La commande BPM vous permet de positionner un point d'arrêt sur la mémoire
en lecture, en écriture, ou en exécution.

Si un verbe n'est pas précisé, c'est RW par défaut.

Si une taille n'est pas précisée, c'est octet par défaut.

Tous les types de verbe sauf X forcent le programme à exécuter l'instruction qui provoque le point d'arrêt. Le CS:IP courant pointera l'instruction qui suit le point d'arrêt. Si le verbe est X, le CS:IP courant pointe l'instruction où le point d'arrêt était positionné.

Si R est précisé, alors le point d'arrêt survient sur un accès en lecture ou en écriture, ces opérations ne changent pas la valeur de l'emplacement mémoire.

Si le type de verbe est R, W ou RW, l'exécution d'une instruction à l'adresse précisée ne provoquera pas l'exécution du point d'arrêt.

Note: Si BPMW est employé, l'adresse précisée doit commencer sur un mot. Si BPMD est employé, l'adresse précisée doit indiquer à un mot double.

Exemple:

```
BPM 1234:SI W EQ 10 C=3
 Cette commande définit un point d'arrêt sur
 un accès mémoire d'un octet. La troisième
```

fois que 10H sera écrit à l'emplacement
1234:SI, le point d'arrêt surviendra.

BPM CS:1235 X
Cette commande définit un point d'arrêt sur
exécution.Le point d'arrêt surviendra la
première fois que l'instruction à adresse
CS:1235 est atteinte. Le CS:IP courant
sera l'instruction où le point d'arrêt
était positionné.

BPMW DS:FOO W EQ M 0XXX XXXX XXXX XXX1
Cette commande définit un point d'arrêt mot
sur les écritures mémoire. Le point d'arrêt
surviendra la première fois que l'emplacement
DS:FOO verra sa valeur écrite avec le bit de
poids fort à 0 et le bit de poids faible à 1.
Les autres bits sont indifférents.

BPM DS:1000 W GT 5
Cet ordre définit un point d'arrêt octet sur
les écritures mémoire. Le point d'arrêt surviendra
la première fois qu'il y aura écriture en DS:1000
d'une valeur supérieure à 5.

BPR

BPR -- Positionne un point d'arrêt sur un domaine mémoire

Syntaxe:

```
    BPR adresse_début adresse_fin [verbe] [C=count]
    adresse_début, adresse_fin -- adresse_début et adresse_fin
                                  définissent le domaine mémoire.

    verbe -- R, W, RW, T ou TW
```

Commentaire:

La commande BPR vous permet de positionner un point d'arrêt sur une zone de mémoire.

Tous les types de verbe sauf T ou TW forcent le programme à exécuter l'instruction qui a causé le point d'arrêt. Le CS:IP courant pointera l'instruction suivant le point d'arrêt.

Il n'y a aucun point d'arrêt domaine sur une exécution. Si un point d'arrêt domaine est désiré sur une exécution, R doit être employé. Le décodage d'une instruction est considéré comme une lecture pour des points d'arrêt domaine.

Si aucun verbe n'est précisé, W sera le verbe par défaut.

Le point d'arrêt domaine dégrade les performances du système dans certaines circonstances. Toute lecture ou écriture dans la page de 4K qui contient le domaine du point d'arrêt est analysée par Soft-ICE. Cette dégradation des performances n'est pas habituellement visible, cependant, elle pourrait être extrême dans des cas d'exception.

Les verbes T et TW activent des domaines de trace arrière sur le domaine
précisée. Ils ne provoquent pas de points d'arrêt, mais notent plutôt des
informations sur les instructions qui peuvent être affichées ultérieurement
avec les commandes SHOW ou TRACE. Pour plus d'information sur les domaines de
trace arrière, voir le chapitre 9.

Exemple:
```
   BPR B000:0 B000:1000 W
```
Cette commande définit un point d'arrêt sur un domaine de mémoire.
Le point d'arrêt surviendra s'il y a la moindre écriture dans la zone
mémoire de l'adaptateur monochrome.

BPIO

BPIO -- point d'arrêt sur accès aux ports I/O

Syntaxe:
```
    BPIO port [verbe] [qualifieur valeur] [C=count]
    port -- Une valeur sur un mot ou un octet
    verbe -- R, W, ou RW
                 R -- Lire (ENTREE)
                 W -- Ecrire (SORTIE)
    qualifieur -- EQ, NE, GT, LT, M
                 EQ -- Egal
                 NE -- Pas Egal
                 GT -- Plus grand Que
                 LT -- Plus petit Que
                 M -- Masque

    la valeur - - Une valeur de mot ou octet
```

Commentaires:

La commande BPIO vous permet de positionner un point d'arrêt sur les
lectures ou les écritures sur les ports I/O.

Si valeur est précisée, elle est comparée avec les données réelles
lues ou écrites par l'instruction IN ou OUT causant le point d'arrêt.
La valeur peut être un octet ou un mot. Si le port I/O est un port 8 bits,
alors les 8 bits de poids faible sont employés dans la comparaison.

Le pointeur d'instruction (CS:IP) pointera sur l'instruction après le IN ou
le OUT qui a causé le point d'arrêt.

Si un verbe n'est pas précisé, RW est la valeur par défaut.

Exemple:
```
  BPIO 21 W NE FF
```

Cette commande positionne un point d'arrêt sur un accès à un port I/O.
Le point d'arrêt surviendra si un des registres de masque du contrôleur
d'interruptions est écrit avec une valeur autre que FFH.

```
  BPIO 3FE R EQ M 11XX XXXX
```

Cette commande définit un point d'arrêt octet sur une lecture d'un port I/O.

Le point d'arrêt surviendra la première fois que le port I/O 3FE sera lu avec une valeur qui a les deux bits de poids fort à 1. Les autres bits peuvent prendre toute valeur.

BPINT

BPINT -- Positionne un point d'arrêt sur une interruption

Syntaxe:

BPINT num-int [<AL | AH | AX >= valeur] [C = compte]
num-int -- Numéro de l'Interruption de 0 à FFH
valeur -- Un octet ou un mot

Commentaire:

La commande BPINT permet d'intervenir dans l'exécution d'une interruption matérielle ou logicielle. En utilisant le registre optionnel AX avec une valeur, des appels DOS ou BIOS spécifiques peuvent être isolés aisément.

Si aucune valeur n'est précisée, un point d'arrêt surviendra quand l'interruption précisée par num-int se produit. Cette interruption peut être une interruption logicielle, matérielle, ou interne.

La valeur optionnelle est comparée avec le registre précisé (AH, AL, ou AX) quand l'interruption survient. Si la valeur correspond au registre précisé, alors le point d'arrêt surviendra.

Quand le point d'arrêt survient, si l'interruption était une interruption matérielle, le CS:IP point sur la première instruction à l'intérieur de la routine d'interruption. La commande INT? peut être employée pour voir où l'exécution en était quand l'interruption est survenue. Si l'interruption était une interruption logicielle, quand le point d'arrêt, le CS:IP pointe sur l'instruction INT ayant causé l'interruption.

Exemple:

BPINT 21 AH=4C

Cette commande positionne un point d'arrêt sur l'interruption 21H Le point d'arrêt surviendra quand la fonction DOS 4CH (termine le programme) est appelée.

BPX

BPX -- Positionne ou supprime un point d'arrêt sur exécution

Syntaxe:

BX [adresse] [C=count]

Commentaire:

La commande BPX vous permet de positionner ou de supprimer un point

d'arrêt
d'exécution dans le source. Quand le curseur est dans la fenêtre de code,
l'adresse n'est pas requise. Le point d'arrêt d'exécution est positionné
à l'adresse du curseur. Si un point d'arrêt d'exécution a déjà été
positionné à l'adresse du curseur, alors le point d'arrêt est
supprimé.

Si la fenêtre de code n'est pas visible ou si le curseur n'est pas dans
la fenêtre de code alors l'adresse doit être précisée. Si l'offset seul
est précisé alors le registre CS sera utilisé comme segment.

Note Technique:
BPX emploie un point d'arrêt style INT3 à moins que l'adresse spécifiée ne
soit en ROM. C'est employé au lieu d'un point d'arrêt registre afin de laisser
plus de points d'arrêt exécution disponibles. Si les circonstances requièrent
l'emploi d'un point d'arrêt registre pour une raison particulière (code pas
encore chargé par exemple) vous pouvez établir un point d'arrêt d'exécution
avec la commande BPM.

Exemple:
```
   BPX.1234
    Cela positionne un point d'arrêt exécution à ligne de source 1234.
```

CSIP

CSIP -- Positionne CS:IP sur un domaine mémoire

Syntaxe:
```
    CSIP [OFF | [NOT] adresse-début adresse-fin]
    NOT -- Quand NOT est précisé, le point d'arrêt ne survient
                     que si le CS:IP est hors du domaine précisée.
    OFF -- Met OFF le test de CS:IP
```

Commentaire:

La commande CSIP force un point d'arrêt à être dépendant de l'emplacement
du pointeur d'instructions quand les conditions de point d'arrêt sont réunies.
Cette fonction est souvent utile quand un programme est soupçonné de modifier
accidentellement du code en-dehors de son domaine.

Quand les conditions de point d'arrêt sont rencontrées, le registre CS:IP
est comparé avec un domaine précis. S'il est dans le domaine, le point
d'arrêt est activé. Pour activer le point d'arrêt quand CS:IP est hors du
domaine, utilisez le paramètre NOT.

Quand un domaine CSIP est précisée, il s'applique à TOUS les points d'arrêt

qui sont actuellement actifs.

Si aucun paramètre n'est précisé, le domaine CSIP courant est affiché.
Exemple:

```
CSIP NE F000:0 FFFF:0
```

Cette commande force le point d'arrêt à survenir seulement le si CS:IP n'est pas dans la ROM BIOS quand les conditions de point d'arrêt sont réunies.

BPAND

BPAND -- Attente du déclenchement de plusieurs points d'arrêt

Syntaxe:

```
   BPAND liste | * | OFF
    liste -- Une série de numéros de points d'arrêt
             séparés par des virgules ou des espaces

    * -- réalise un AND entre tous les points d'arrêt
```

Commentaire:

La commande BPAND fait un ET logique de deux ou plus points d'arrêt, en ne les activant que quand les conditions de tous les points d'arrêt sont remplies.

Parfois, les conditions surviennent quand vous ne voulez pas qu'un point d'arrêt soit activé jusqu'à ce que plusieurs conditions différentes soient rencontrées. La commande BPAND permet de préciser deux ou plusieurs points d'arrêt qui doivent survenir avant que l'action se déclenche.
Cette fonction permet de définir des conditions de point d'arrêt plus complexes.

Chaque fois que la commande BPAND est employée, les numéros de points d'arrêt précisés sont ajoutés à la liste jusqu'à ce que BPAND OFF soit employé.

Vous pouvez savoir quels points d'arrêt sont liés avec la commande AND en listant les points d'arrêt avec la commande BL. Les points d'arrêt liés par la commande AND auront un ampersand (&) après leur numéro.

Une fois que des points d'arrêt ont été liés avec AND, chacun reste lié par AND jusqu'à ce qu'il soit effacé, ou jusqu'à ce que BPAND soit mis OFF.

Exemple:
BPAND 0,2,3

Cette commande force les conditions des points d'arrêt 0, 2, et 3 à être logiquement liées ensemble. Le point d'arrêt survient seulement quand les conditions de tous les trois sont réunies. Par exemple, si les

conditions des points d'arrêt 2 et 3 ont toutes les deux été rencontrées
au moins une fois chacune, mais que les conditions du point d'arrêt 0 n'ont pas encore été rencontrées, alors l'action ne surviendra pas tant
que les conditions du point d'arrêt 0 ne seront pas rencontrées.

4.3 Manipulation des points d'arrêt

Soft-ICE fournit plusieurs commandes pour manipuler des points d'arrêt.
Les commandes de manipulation permettent de lister, de modifier, d'effacer,
d'activer, et de désactiver des points d'arrêt. Les points d'arrêt sont
identifiés par leurs numéros qui sont des chiffres hexadécimaux de 0 à F.
Les commandes de manipulation de point d'arrêt sont:

BD -- Désactiver des points d'arrêt
BE -- Activer des points d'arrêt
BL -- Lister des points d'arrêt
BPE -- Editer des points d'arrêt
BPT --Utiliser un point d'arrêt comme modèle
BC -- Effacer des points d'arrêt
BD -- Désactiver des points d'arrêt

BD

Syntaxe:
```
   BD liste | *
   liste -- Une série numéros de points d'arrêt séparés
                 par des virgules ou des espaces

   * -- Désactive tous les points d'arrêt
```

Commentaire:

La commande BD est utilisé pour désactiver temporairement des points d'arrêt.
Les points d'arrêt peuvent être réctivés avec la commande BE (Activer point
d'arrêt).

Vous pouvez préciser quels points d'arrêt sont désactivés en énumérant les
points d'arrêt avec la commande BL. Les points d'arrêt désactivés sont repérés
avec un astérisque (*) après leur numéro.

Exemple:
```
   BD 1,3

   Cette commande désactive temporairement les points
         d'arrêt 1 et 3.
```

BE

BE -- Active des points d'arrêt

Syntaxe:
```
    BE liste | *
```

liste -- Une série de numéros de points d'arrêt séparés
par des virgules ou des espaces

* -- Active tous les points d'arrêt

Commentaire:

La commande BE est utilisée pour réactiver des points d'arrêt qui avaient
été désactivés par la commande BD.

Notez qu'un point d'arrêt est automatiquement activé quand il est défini pour la première fois.

Exemple:
BE 3

Cette commande active le point d'arrêt n°3.

BL

BL -- liste les points d'arrêt

Syntaxe:
BL

Commentaires:

La commande BL affiche tous les points d'arrêt actuellement positionnés.Pour
chaque point d'arrêt, BL énumère le numéro, les conditions, l'état, et le
compteur.

L'état d'un point d'arrêt est soit activé, soit désactivé. Si le point d'arrêt
est désactivé, un astérisque (*) est affiché après son numéro. Si un point
d'arrêt activé est employé dans une commande BPAND, un ampersand (&) est affiché
après son numéro. Le point d'arrêt qui a causé une action le plus récemment
est affiché en surbrillance..

La commande BL n'a pas de paramètres.

Exemple:
BL

Cette commande affiche tous les points d'arrêt qui ont été définis.
Voici un exemple qui affiche 4 points d'arrêt:

```
0) BPMB 1234:0000 W EQ 0010 C=03
1) * BPR B000:0000 B000:1000 W C=01
2) BPIO 0021 W NE 00FF C=01
3) BPINT 21 AH=4C C=01
```

Noter que dans cet exemple, le point d'arrêt 1 est précédé d'un astérisque (*), montrant qu'il a été dévalidé.

BPE

BPE -- Edite un point d'arrêt

Syntaxe:
   BPE numéro du point d'arrêt

Commentaire:

La commande BPE charge la description du point d'arrêt sur la ligne de
commande pour la modification. La commande peut alors être éditée en utilisant les touches d'édition et revalidée en pressant ENTREE. Cette
commande offre une méthode rapide pour modifier les paramètres d'un point
d'arrêt existant.

Exemple:
  BPE 1

   Cette commande déplace la description du
   point d'arrêt 1 dans la ligne d'édition et
   efface le point d'arrêt 1. Presser la touche
   ENTREE réactivera le point d'arrêt avec ses
   nouveaux paramètres s'il y a lieu.

BPT

BPT -- Utilise un point d'arrêt comme modèle

Syntaxe:
   BT numéro du point d'arrêt

Commentaire:

La commande BPT emploie une description de point d'arrêt comme modèle pour un nouveau point d'arrêt.

Une description du point d'arrêt existant est chargée sur la ligne d'édition.
Le point d'arrêt référencé par le numéro n'est pas modifié. Cette commande
offre une méthode rapide pour créer un nouveau point d'arrêt similaire à un
déjà existant.

Exemple:
  BPT 3

   Cette commande déplace un modèle du point d'arrêt 3
   sur la ligne d'édition. Quand la touche ENTREE est
   pressée, un nouveau point d'arrêt est ajouté.

BC

BC -- Efface tous les points d'arrêt

Syntaxe:

```
BC liste | *
liste -- Une série de numéros de points d'arrêt
         séparés par des virgules ou des espaces

* -- Efface tous les points d'arrêt
```

Commentaire:

La commande BC est employée pour effacer définitivement un ou plusieurs
points d'arrêt.

Exemple:

```
BC *
```

Cette commande efface tous les points d'arrêt.

CHAPITRE 5

Utilisation d'autres Commandes

5.1 Commandes d'édition et d'affichage

```
Commandes:
U        -- Désassemble instructions ou affiche un source
R        -- Affiche ou modifie des registres
MAP      -- Affiche une carte mémoire du système
D        -- Affiche la mémoire dans le dernier format précisé
DB       -- Affiche la mémoire dans le format octet
DW       -- Affiche la mémoire dans le format mot
DD       -- Affiche la mémoire dans le format mot double
E        -- Edite la mémoire dans le dernier format précisé
EB       -- Edite des octets en mémoire
EW       -- Edite des mots en mémoire
ED       -- Edite des mots doubles en mémoire
INT?     -- Affiche le dernier numéro d'interruption
? ou H   -- Affiche des Informations d'Aide
VER      -- Affiche le numéro de version de Soft-ICE
```

U

U -- Désassemble des instructions ou affiche un source

Syntaxe:

    U [adresse] [L[=]longueur]

   longueur -- Le nombre d'instructions à désassembler

Commentaires:

La commande U affiche les instructions du programme en débugging.

Si longueur n'est pas précisée, la longueur par défaut est fixée à huit
lignes s'il y en a huit disponibles, ou une de moins que la longueur d'écran.

Si adresse n'est pas précisée, la commande désassemble à l'adresse commençant
au premier octet après le dernier octet désassemblé par un précédent désassemblage. S'il n'y a eu aucun précédent désassemblage, l'adresse par
défaut sera fixée au CS:IP.

Si la fenêtre de code est visible, les instructions sont affichées dans la fenêtre de code.

Si un source est chargé pour le domaine d'adresses précisé, alors des lignes de source peuvent être affichées en accord avec le mode SOURCE actuel.

Exemple:

```
U $-10
```

Cette commande désassemble des instructions en commençant 10H octets avant l'adresse actuelle.

```
U .499
```

Cette commande affiche le fichier source actuel en commençant à la ligne 499. La fenêtre de code doit être visible et en mode SOURCE.

R

R -- Affichage ou modifications de registres

Syntaxe:

```
 R nom-registre [ [ = ] valeur]

 nom-registre -- Un des noms suivants:
                  AL, AH, AX, BL, BH,
                  BX, CL, CH, CX, DL,
                  DH, DX, DI, SI, BP,
                  SP, IP, CS, DS, ES,
                  SS, ou FL
 valeur - - Si nom-registre est différent de FL,
                  valeur est une valeur hexa ou une
                  expression. Si nom-registre est FL,
                  valeur est une série d'un ou plusieurs
                  des drapeaux suivants, chacun précédé
                  de façon optionnelle par un signe plus
                  ou moins:
                     O (drapeau de débordement)
                     D (drapeau de direction)
                     I (drapeau d'Interruption)
                     S (drapeau de signe)
                     Z (drapeau de zéro)
                     A (drapeau Auxiliaire de Retenue )
                     P (drapeau de Parité)
                     C (drapeau de Retenue)
```

Commentaires:
La commande R modifie ou affiche la valeur des registres.

Si aucun paramètre n'est fourni, les valeurs des drapeaux et des registres sont affichées, ainsi que l'instruction pointée par CS:IP.

Si nom-registre est entré sans valeur, Soft-ICE affiche la valeur actuelle du registre cité etattend une nouvelle valeur. Si nom-registre est FL, les drapeaux positionnés sont affichés en majuscules et en surbrillance; les drapeaux non positionnés sont affichés en minuscules et brillance normale. Pour conserver la valeur actuelle d'un registre, pressez ENTREE.

Si nom-registre et valeur sont précisés, les contenus des registres spécifiés
sont modifiés.

Pour changer d'un valeur d'un drapeau, utilisez FL comme nom-registre, suivi
par les symboles des drapeaux dont vous voulez changer la valeur. Pour mettre
un drapeau ON, faire précéder le symbole de ce drapeau par un signe plus. Pour
mettre un drapeau OFF, faire précéder le symbole de ce drapeau par un signe
moins. Les drapeaux peuvent être énumérés dans nimporte quel ordre.

Exemples:

```
R AH 5
```

Cette commande positioonne le registre AH à 5.

```
R FL = OZP
```

Cette commande bascule la valeur des drapeaux O, Z, et P.

```
R  FL
```

Cette commande affiche les valeurs actuelles des drapeaux,
et permet leur changement.

```
R FL O +A -C
```

Cette commande inverse la valeur du drapeau O, positionne
la valeur du drapeau A sur 1 et celle du drapeau C sur 0.

MAP

MAP -- Affiche une carte de la mémoire système

Syntaxe:

MAP

Commentaires: La commande CARTE affiche les noms, les emplacements et les
tailles des composants de la mémoire système. La taille est exprimée en
paragraphes. Un paragraphe est équivalent à 10H octets.

Le composant pointé par le CS:IP est affiché en surbrillance.

Employer la commande MAP quand:

* Un point d'arrêt survient et CS:IP n'est pas dans une région connue de la mémoire.
* Vous voulez obtenir le contrôle d'un programme résidant ou d'un programme système. Un point d'arrêt domaine peut être positionné basé sur la taille et l'adresse de début affichée par MAP.
* Vous soupçonnez un composant du système ou un programme d'écrire sur le code hors de son espace mémoire. MAP est employée obtenir l'adresse de mémoire de la région à utiliser avec la commande CSIP.
* Vous souhaitez trouver quel programme résidant

intercepte certains vecteurs d'interruption.

Exemple:

MAP

Voici un exemple d'affichage produit par la commande:

```
Start     Length
0000:0000 0040 Table des vecteurs d'Interruption
0040:0000 0030 ROM BIOS Variables
0070:0000 00FE I/O Système
016E:0000 06B7 DOS
0842:0000 02CE DOS Table Des fichiers & Buffers
A000:0000 5E00 BUS système
F000:0000 1000 ROM BIOS
```

Les versions de DOS inférieures à 3.1 affichent les adresses des programmes au lieu de leurs noms.

D, DB, DW, DD

D, DB, DW, DD -- Affiche la mémoire

Syntaxe:

```
D [taille] [adresse] [L[ = ]longueur]
taille -- B -- Octet
          W -- Mot
          D -- Mot Double

longueur -- Le nombre d'octets à afficher.
```

Commentaires:

La commande D affiche le contenu de la mémoire à l'adresse spécifiée.

Les contenus sont affichés dans le format de taille précisé. Si aucune taille n'est précisée, la dernière taille employée sera utilisée. La représentation ASCII est aussi utilisée pour tous les formats.

Si adresse n'est pas précisée, la commande affiche le contenu de la mémoire à l'adresse commençant au premier octet après le dernier octet affiché la fois précédente.

Si longueur n'est pas précisée, elle est fixée par défaut à huit lignes, ou moins si la fenêtre est plus petite.

Si la fenêtre de données est visible, les données sont affichées dans la fenêtre de données et la longueur est ignorée.

Exemple:

DW DS:00 L=8

Cette commande affiche, dans le format de mot et dans le format ASCII, la valeur des huit premiers octets du segment de données actuel.

E, EB, EW, ED

E, EB, EW, ED -- Edite la mémoire

Syntaxe:

```
    E [taille] adresse [liste-données]
    taille -- B -- Octet
              W -- Mot
              D -- Mot Double

    liste-données -- liste d'objets de données de la
                        taille précisée (Octets, Mots
                        ou Mots Doubles) ou de chaînes
                        séparées par des virgules ou
                        des espaces. Les chaînes peuvent
                        commencer avec une quote simple ou
                        une quote double (' ou ").
```

Commentaire:
La commande E affiche le contenu de la mémoire à l'adresse précisée, et vous
permet d'éditer les valeurs.

Ces commandes affichent le contenu de la mémoire dans le format ASCII, et
dans le format de taille précisé.

Un éditeur de mémoire est fourni pour de rapides modifications de mémoire.
La mémoire peut être éditée en entrant des caractères ASCII ou en entrant les
valeurs d'octet, de mot, ou de mot double. Si aucune taille n'est précisée, la
dernière taille employée sera supposée. Les touches d'édition mémoire sont:

```
    _ -- Déplacement du curseur vers le haut
    _ -- Déplacement du curseur vers le bas
    _ -- Déplacement du curseur vers la droite
    _ -- Déplacement du curseur vers la gauche
    ESPACE -- Déplacement du curseur vers l'élément suivant
    TAB -- Bascule entre la zone numérique et la zone ASCII
    ESC ou
    ENTER -- Sortie de l'éditeur de mémoire
```

Toutes les valeurs sont validées, les emplacements mémoire sont mis à jour.
Toutes les valeurs numériques sont exprimées en hexadécimal. Pour basculer
entre la zone ASCII la zone numérique, utilisez la touche TAB.

Si la fenêtre de données est visible, les données sont éditées dans cette
fenêtre, sinon les données sont éditées dans la fenêtre de commandes.

Les données sont par défaut sur une longueur de 8 lignes dans la fenêtre de
de commandes, ou adaptées à la taille de la fenêtre de données si elle est
visible.

Si aucun paramètre n'est fourni, le curseur est déplacé dans la fenêtre de
données si elle est visible. Si la fenêtre de données n'est pas visible, les
données sont éditées dans la fenêtre de commandes à la dernière

adresse affichée ou éditée.

Exemples:

EB 1000:0

Cette commande affiche, dans le format d'octet, jusqu' à six lignes contenant les représentations numérique et ASCII des valeurs des données commençant à l'emplacement 1000:0000. Une fois les lignes affichées, vous pouvez éditer les valeurs.

EB 8000:0 "Bonjour" , 0D

Cette commande remplace les valeurs commençant à l'emplacement 8000:0000 par la chaine "Bonjour" suivie par un retour chariot.

INT?

INT? -- Affiche le numéro de la dernière interruption

Syntaxe:

INT?

Commentaire:
La commande INT? affiche l'adresse et le numéro de la dernière interruption qui s'est produite.

Exemple:

INT?

Un exemple de l'affichage produit par la commande INT?:

```
Last Interrupt: 16
At: 0070:0255
```

Cet exemple montre que la dernière interruption générée dans le système avant que Soft-ICE soit activé était l'interruption 16H, à l'adresse 0070:0255H. Si la dernière interruption qui est survenue était une interruption logicielle, le désassemblage du code à l'adresse 0070:0255H montrera l'instruction d'interruption. Si c'était une interruption matérielle, le désassemblage du code montrera l'instruction qui s'exécutait quand l'interruption matérielle est survenue.

? ou H

? ou H -- Affiche des informations d'aide

Syntaxe:

< ? | H > [commande | expression]

Commentaires:
La commande ? et La commande H affichent toutes les deux les informations d'aide.

Si aucun paramètre n'est précisé, l'aide affiche une courte description de toutes les commandes et opérateurs, un écran à la fois. Presser une touche pour continuer, ou ESC pour abandonner l'aide.

Si commande est précisé, l'aide affiche une information plus détaillée sur
la commande précisée, incluant la syntaxe de la commande et un exemple.

Si expression est précisée, l'expression est évaluée et le résultat est
affiché en hexadécimal, décimal, et ASCII.

Exemples:
```
  ? ALTKEY
    Cette commande affiche une l'information sur
    la commande ALTKEY, incluant sa syntaxe et un
    exemple.
  H 10 + 14*2
    Cette commande affiche: 0038 00056 "8". Ce sont
    les représentations hexadécimales, ASCII et
    décimales de la valeur de l'expression "10 + 14 * 2".
```

VER

VER -- affiche le numéro de version de Soft-ICE

Syntaxe:

```
  VER
```

Exemple:
```
   VER
   Cette commande affiche la version de Soft-ICE
   et le message de copyright de Nu - Mega Technologies.
```

5.2 Commandes des Ports I/O

Commande:
```
   I ou IB -- Entrée d'1 octet depuis un port I/O
   IW -- entrée d'un mot depuis un port I/O
   O ou OB -- Sortie d'un octet sur un port I/O
   OW -- Sortie d'un mot sur un port I/O
```

I, IB, IW

I, IB, IW -- Entrée depuis un port I/O port

Syntaxe:

```
   I [taille] port

   taille -- B -- Octet
             W -- Mot

   port -- Une valeur de mot ou d'octet
```

Commentaire:
La commande d'entrée depuis un port est employée pour lire et afficher
une valeur d'un port matériel. L'entrée peut être faite en mot ou en
octet. Si aucune taille n'est précisée, le défaut est octet.

Exemple:
```
   I 21
   Cette commande affiche le registre de masque
```

du contrôleur d'interruptions un.

O, OB, OW

O, OB, OW, -- Sortie sur un port I/O

Syntaxe:

```
O [taille] valeur du port
taille -- B -- Octet
          W - - Mot

port -- valeur de mot ou d'octet
valeur -- Un octet pour un port d'octet
          ou un mot pour un port de mot
```

Commentaires:
Les commandes de sortie sur un port sont employés écrire une valeur vers
un port matériel. La sortie peut être faite sur un port de mot ou d'octet.
Si aucune taille n'est précisée, le défaut est octet.

Exemple:
O 21 FF
Cette commande masque toutes les interruptions
du contrôleur d'interruptions un.

5.3 Commandes du Contrôle De transfert

Commandes:

X -- Sortie de la fenêtre Soft-ICE
G -- Aller à une adresse
T -- Trace une instruction
P -- Pas de programme
HERE -- Aller à la ligne du curseur
GENINT -- Force une interruption
EXIT -- Force la sortie du programme DOS courant
BOOT -- Boot du système (retient Soft-ICE)
HBOOT -- Boot dur du système (reset total)

X

X -- Sortie de la fenêtre Soft-ICE

Syntaxe:

X

Commentaires:
La commande X quitte la fenêtre Soft-ICE et rend le contrôle au programme qui
avait été interrompu pour activer Soft-ICE. La fenêtre Soft-ICE disparaît.
Si un point d'arrêt a été positionné, il reste actif.

Exemple:
X

G

G -- Aller à l'adresse

Syntaxe:

    G [=adresse-départ] [adresse-arrêt]

Commentaires:
La commande G quitte la fenêtre soft-ICE avec un point d'arrêt à exécution
unique positionné. De plus, tous les points d'arrêts permanents du programme
restent validés.

L'exécution commence au CS:IP à moins que le paramètre adresse-départ soit
précisé. Dans ce cas, l'exécution commence à adresse-départ. L'exécution
continue jusqu'à ce que adresse-arrêt soit atteinte, que la séquence de touches
de sortie soit employée ou qu'un point d'arrêt permanent survienne.

Adresse-arrêt doit être le premier octet d'une instruction.

Quand l'adresse-arrêt spécifiée est atteinte, le CS:IP pointera sur
l'instruction où le point d'arrêt était positionné.

La commande G sans paramètre se comporte comme la commande X.

Le point d'arrêt d'exécution temporaire utilise un point d'arrêt registre
du 80386, à moins que tous les point d'arrêt registre aient été affectés
aux points d'arrêt permanents. Dans ce cas, un point d'arrêt de style INT3
est utilisé. Quand ce cas survient, les commandes G et P ne fonctionneront
pas correctement dans la ROM. Un message d'erreur sera affiché.

Exemple:
   G CS:1234
    Cette commande positionne un point d'arrêt à utilisation
    unique à CS:1234

T

T -- Trace une instruction

Syntaxe:

    T [=start-adresse] [compte]

Commentaires:

Les commandes de type T passent une instruction après l'autre en utilisant
le drapeau pas à pas.

L'exécution commence au CS:IP à moins que le paramètre start-adresse soit
précisé. Si start-adresse est précisé, CS:IP est modifié en start-adresse
avant la trace.

Si compte est précisé la commande TRACE sera exécutée jusqu'à ce que

le compte
soit épuisé ou que la touche ESC soit pressée en tenant compte bien sûr
d'éventuels points d'arrêt permanents du programme.

En mode source, la commande T saute à la prochaine ligne de source. Si
la ligne actuelle est un appel de fonction ou de procédure, et si le source
existe pour la routine appelée, T continue dans la routine. S'il n'y a pas de
source disponible pour la fonction ou la procédure appelée, T saute par-dessus
la routine.

Exemple:

```
   T = 1284 3
    Cette commande exécute une par une trois instructions
    commençant à l'emplacement mémoire 1284.
```

P

P -- Exécution de Programme

Syntaxe:

```
   P
```

Commentaires:
La commande P est un saut logique dans un programme. Une instruction pointée
par le CS:IP est exécutée à moins que l'instruction ne soit un appel, une
interruption, une boucle, ou une instruction de répétition. Dans ces cas, la
routine entière ou l'itération est entièrement exécutée avant de rendre le
contrôle à soft-ICE.

La commande P emploie un point d'arrêt unique d'exécution.Le point d'arrêt
d'exécution non permanent emploie un point d'arrêt registre du 80386, à moins
que tous les points d'arrêt registre aient été affectés aux points d'arrêts
permanents. Dans ce cas, un point d'arrêt de style INT3 est réalisé. Quand ce
cas survient, les commandes P et G ne fonctionneront pas correctement dans la
ROM. Un message d'erreur sera affiché si celà est tenté.

Dans le mode source, la commande P exécute jusqu'à la prochaine ligne de source.
Si la ligne actuelle est un appel de fonction ou de procédure, la commande P
saute par-dessus.

Exemple:

```
    P
    Cette commande exécute un pas de programme.
```

HERE

HERE -- Aller a la ligne du curseur

Syntaxe:

  HERE

Commentaires:
La commande HERE exécute jusqu'à ce que le programme atteigne la ligne du curseur. HERE est seulement disponible quand le curseur est dans la fenêtre de code. Si la fenêtre de code n'est pas visible ou si le curseur n'est pas dans la fenêtre de code, employez plutôt la commande G.

La commande HERE sort de Soft-ICE avec un point d'arrêt d'exécution positionné.
En plus, tous les points d'arrêt permanents sont armés.

L'exécution commence au CS:IP et continue jusqu'à ce que l'adresse du curseur dans la fenêtre de code soit rencontrée, que la séquence de touche de sortie de la fenêtre soit activée, ou qu'un point d'arrêt permanent survienne.

Le point d'arrêt d'exécution temporaire emploie un point d'arrêt registre du 80386 , à moins que tous les point d'arrêt registre aient été affectés aux points d'arrêt permanents. Dans ce cas, un point d'arrêt de style INT3 est réalisé. Quand ce cas se produit, la commande HERE ne fonctionnera pas correctement dans la ROM. Un message d'erreur sera affiché si celà est tenté.

Exemple:
```
 HERE
    Cet exemple positionne un point d'arrêt d'exécution
    à la position courante du curseur, puis quitte Soft-ICE
    et commence l'exécution au CS:IP.
```
Touche de fonction par défaut: F7

GENINT

GENINT -- Force une interruption

Syntaxe:

   GENINT INT1 | INT3 | NMI | num-interruption

   num-interruption -- un nombre dans la gamme 00 - FF

Commentaires:
La commande GENINT force une interruption à survenir. Cette fonction peut être employée pour donner le contrôle à un autre débugger si vous utilisez Soft-ICE avec un autre débugger. Elle peut aussi être employée pour tester des routines d'interruption.

La commande GENINT simule la séquence de traitement d'une interruption matérielle ou d'une instruction INT. Elle pousse sur la pile: les drapeaux, le

registre CS et le registre IP, elle change alors la valeur de CS et de IP
par les valeurs correspondantes dans la table des vecteurs d'interruption avec
le numéro d'interruption précisé.

Exemple:
    GENINT NMI
    Cela force une interruption non-masquable.
    Cela donnera le contrôle à CodeView si
    Soft-ICE est employé comme un assistant à
    CodeView.

EXIT

EXIT -- Force la sortie du programme DOS actuel

Syntaxe:

```
   EXIT [R] [D]
    R -- Restaurer la table des vecteurs d'interruption
    D -- Effacer tous les points d'arrêt
```

Commentaires:
La commande EXIT essaie de terminer le programme actuel en forçant une fonction
DOS de sortie (INT 21H, fonction 4CH). Cette commande ne fonctionnera que si
le DOS est dans un état où il est capable d'accepter l'appel de cette fonction
de sortie. Si cet appel est fait depuis certaines routines d'interruption, ou
quand le DOS n'est pas prêt, le système peut se comporter de façon imprévisible.

Cette fonction NE remet aucun système à zéro sinon les tables d'interruption
quand l'option R est employée. Cela signifie que variables BIOS, mode vidéo et
autres données systèmes ne sont pas restaurés.

Utiliser l'option R restaurera les vecteurs d'interruption à leur dernière
valeur sauvegardée. Soft-ICE sauvegarde les vecteurs d'interruption quand il
est chargé, quand un programme est chargé avec LDR.EXE, et quand la commande
VECS S est employée.

Notes:
Pour redémarrer un programme qui a été chargé avec LDR.EXE procéder ainsi:

```
EXIT R
LDR prog.EXE
```

La commande EXIT restaurera la table d'interruptions aux valeurs qu'elle
contenait avant que le programme ne soit chargé, puis quittera le programme.

Avec l'utilitaire LDR et en précisant le suffixe .EXE, le programme est chargé
sans rechargement des symboles et du source. Les symboles et le source restent
en mémoire.

Prudence:
La commande EXIT doit être employée avec précautions. Malgrès que Soft-ICE puisse être activé à tout moment, une situation peut survenir où le DOS n'est pas en état d'accepter un appel de fonction de sortie. Aussi, la commande EXIT ne peut pas remettre à zéro tout programme. Par exemple, la commande EXIT ne remet pas à zéro le mode vidéo. Si votre programme a placé le BIOS vidéo dans un mode particulier, il restera dans ce mode après la commande EXIT.

Exemple:
  EXIT R
    Restaure la table d'interruption et sort du programme
    actuel. L'option R doit être employée pour sortir d'un
    programme chargé avec LDR.EXE.

BOOT

BOOT -- BOOTE le système (laisse Soft-ICE en mémoire)

La syntaxe:

BOOT

Commentaires:
La commande BOOT remet le système à zéro et garde Soft-ICE en mémoire. BOOT est nécessaire pour débugger des séquences de démarrage, des pilotes de périphériques, et des systèmes d'exploitation non-DOS.

BOOT est implémenté grâce à un appel à l'interruption 19H de la ROM BIOS. Dans certaines circonstances, la mémoire peut être corrompue au point que l'interruption 19 ne fonctionne pas. Si cela survient, activez Soft-ICE et utilisez la commade HBOOT.

Pour que BOOT puisse fonctionner correctement, Soft-ICE doit être installé comme un pilote dans CONFIG.SYS avant tout autre pilote. Ceci pour que Soft-ICE puisse restaurer l'état original du système aussi exactement que possible.

Exemple:
  BOOT
   Cette commande fait le rebooter le système. Soft-ICE
   reste résident.

HBOOT

HBOOT -- BOOT hard du système (reset total)

Syntaxe:

HBOOT

Commentaires:
La commande HBOOT remet le système intégralement à zéro. Soft-ICE ne reste
pas en mémoire. HBOOT est normalement suffisant à moins qu'une carte d'extention ne requière un reset 'électrique'. Dans ces cas rares, l'alimentation de la machine doit être coupée.

Exemple:
  HBOOT
   Cette commande reboote totalement le système.
   Soft-ICE doit être rechargé.

5.4 Commandes du Mode Debug

Commande:
   ACTION -- Définit l'action après qu'un point d'arrêt soit atteint
   WARN   -- Positionne le mode d'avertissement de
            réentrance DOS/ROM BIOS
   BREAK -- Arrêt à tout moment
   I3HERE -- Dirige l'Interruption 3 vers Soft-ICE

ACTION

ACTION -- Définit l'action après qu'un point d'arrêt soit atteint

Syntaxe:

   ACTION [INT1 | INT3 | NMI | HERE | num-int]

   num-int -- Tout numéro d'interruption valable (0-FFH).
              N'employer cette option que si une routine
              utilisateur a détourné le vecteur (voir la section
11.2).

Commentaires:
La commande ACTION détermine à qui est donné le contrôle quand les conditions
de point d'arrêt ont été rencontrées. Dans la plupart des cas, l'action
désirée est INT3 ou HERE, INT3 est typiquement employé si Soft-ICE est
employé avec un débugger hôte, HERE est employé quand on souhaite revenir
à Soft-ICE quand les conditions de point d'arrêt ont été rencontrées, INT1 et
NMI sont des alternatives pour certains débuggers qui ne travaillent pas avec
l'option INT3. Par exemple, CODEVIEW travaille mieux avec ACTION positionné
sur NMI.

N'employez num-int que s'il y a une routine de point d'arrêt utilisateur
installée. Utiliser num-int en l'asence d'une telle routine provoque une
erreur. Pour plus d'information, voir la section 11.2, 'Points d'arrêt
utilisateur'.

Si aucun paramètre n'est fourni avec la commande ACTION, la valeur

actuelle
d'action est affichée.

L'action par défaut est HERE.

Exemple:
  ACTION HERE
Cette commande précise que le contrôle reviendra à Soft-ICE quand les conditions de point d'arrêt auront été rencontrées.

WARN

WARN -- Positionne l'avertissement de réentrance DOS/ROM BIOS

Syntaxe:

  WARN [ON | OFF]

Commentaires:
La commande WARN est fournie pour utiliser Soft-ICE avec des débuggers qui
emploient le DOS et la ROM BIOS. Beaucoup de débuggers utilisent le DOS et
la ROM BIOS pour les sorties écran et pour la saisie du clavier. Comme le DOS
et la ROM BIOS ne sont pas pleinement réentrants, ces débuggers ne peuvent pas
travailler correctement si un point d'arrêt survient tandis que le DOS ou la
ROM BIOS travaillent.
Si WARN est positionné sur ON, et si ACTION n'est pas HERE, alors le contrôle
sera donné à Soft-ICE avant que l'action réelle ne survient. Le système affiche
le CS:IP et vous donne le choix entre continuer ou revenir à Soft-ICE.
Généralement, vous devriez choisir de revenir à Soft-ICE pour continuer votre
débugging. Ne continuez avec le débugger hôte que si vous savez que votre
débugger ne forcera pas DOS ou la ROM BIOS à réentrer.

Le mode WARN devrait être sur ON pour employer Soft-ICE avec DEBUG, SYMDEB,
et CODEVIEW. Si aucun paramètre n'est précisé, l'état actuel de WARN est
affiché. Le mode par défaut de WARN est OFF.

Exemple:
  WARN ON
   Cette commande met sur ON l'avertissement de réentrance DOS/ROM BIOS.

BREAK

BREAK -- Arrête à nimporte quel moment

Syntaxe:

BREAK [ON | OFF]

Commentaires:
La commande BREAK permet d'activer la fenêtre Soft-ICE quand le système est

planté avec les interruptions désactivées. Le mode BREAK peut être employée
pour la séance de débugging entière, ou il peut être mis ON et OFF quand cela
est requis.

Le mode BREAK dégrade légèrement les performances de votre système. Cette
dégradation de performances doit être mise en bascule avec la nécessité de
sortir d'un programme planté. Un utilisateur peut vouloir avoir le mode BREAK
ON tout le temps, bien que les performances soint dégradées, parce que le
programme pourrait le suspendre à tout moment.

A la différence d'autres débuggers qui peuvent aussi être activés à tout
moment, Soft-ICE ne requiert pas de commutateur externe. Quand BREAK est on,
la fenêtre Soft-ICE peut être activée à tout moment en pressant la séquence
actuelle de touches.

Si aucun paramètre n'est précisé, l'état actuel de BREAK est affiché. Le mode
de BREAK par défaut est OFF.

Exemple:

```
  BREAK ON
    Cette commande met le mode BREAK ON.
    Cela signifie que la fenêtre Soft-ICE
    peut être activée a tout moment, même
    si les interruptions sont désactivées.
```

13HERE

13HERE -- Dirige l'nterruption 3 sur Soft-ICE

Syntaxe:

```
  13HERE [ON | OFF]
```

Commentaires:
La commande 13HERE vous permet de préciser qu'aucune Interruption3 n'activera
la fenêtre Soft-ICE. Cette caractéristique est utile pour arrêter votre
programme dans un emplacement mémoire spécifique.

Pour employer cette caractéristique, placez un INT3 dans votre code à l'endroit
où vous voulez l'arrêter. Quand l'INT3 survient, elle activera la fenêtre
Soft-ICE. A ce point, vous pouvez utiliser la commande R IP pour changer votre
pointeur d'instruction à l'instruction suivant l'INT3, alors vous pouvez
continuer à débugger.

Si aucun paramètre n'est précisé, l'état actuel de 13HERE est affiché.

Le mode par défaut de 13HERE OFF.

```
Exemple:
  13HERE ON
    Cette commande met le mode 13HERE sur ON.
    Toute INT3 générée après ce point
    activera la fenêtre Soft-ICE.
```

5.5 Commandes Utilitaires

```
Commandes:
    A     -- Assembler le code
    S     -- Rechercher des données
    F     -- Remplir la mémoire avec des données
    M     -- Déplacer des données
    C     -- Comparer deux blocs de données
```

A

A -- Assembler le code

Syntaxe:

    A [adresse]

Commentaires:
L'assembleur de Soft-ICE vous permet d'assembler des instructions directement
en mémoire. L'assembleur supporte les instructions fondamentales du jeu 8086
avec les extensions d'adressage en mode réel des 80186 et 80286. Les instructions des coprocesseurs arithmétiques et les instructions 80386,
registres et modes d'adressage spécifiques NE peuvent pas être assemblées.

La commande A entre dans l'assembleur interactif de Soft-ICE. Une adresse est
affichée comme prompt pour chaque ligne assemblée. Après qu'une instruction
d'assembleur soit entrée et que ENTREE soit pressée, les instructions sont assemblées en mémoire à l'adresse précisée. Les instructions doivent
être entrées au format Intel standard. Pressez ENTREE à un prompt pour sortir
de l'assembleur.

Si la zone d'adresses dans laquelle vous assemblez des instructions est visible
dans la fenêtre de code, les instructions changeront de façon interactive au
fur et à mesure de l'assemblage.

L'assembleur de Soft-ICE supporte les mnémoniques standards de la famille
8086, cependant il y a certains ajouts:
   * La mnémonique DB est utilisée pour définir des
       octets de données directement en mémoire. La
       commande DB est suivie par une liste d'octets
       et/ou des chaînes séparées par des espaces ou
       des virgules et encadrées de guillemets.
   * La mnémonique RETF représente un retour lointain.
   * WORD PTR et BYTE PTR sont employés pour préciser

la taille des données à évaluer s'il n'y a
aucun registre dans l'argument, par exemple:
MOV BYTE PTR ES:[1234],1.

* Utiliser FAR et NEAR pour assembler explicitement
les saut et les appels proches et lointain.
Si FAR ou NEAR ne sont pas précisés, alors
tous les sauts et appels sont assemblés en
proches.

* Les Opérandes se référant aux emplacements de
mémoire doivent être placés entre des crochets,
par exemple: MOV AX,[1234].

Exemple:
```
A CS:1234
 Cette commande attend l'entrée d'une instruction
 assembleur puis l'assemble en commençant à l'offset
 1234H dans le segment de code actuel. Pressez ENTREE
 au prompt après avoir entré la dernière instruction.
```

S

S -- Recherche de données

Syntaxe:

```
S adresse L longueur liste-données
liste-données -- liste d'octets ou chaînes séparées par
                    des virgules ou des espaces et encadrées
                    par des guillemets. Une chaîne peut
                    commencer avec une apostrophe seule ou
                    un guillemet.

longueur - - longueur en octets
```

Commentaires:

La commande S recherche dans la mémoire une série d'octets ou de caractères
correspondant à liste-données. La recherche commence à l'adresse précisée
et continue sur la longueur précisée. L'adresse de chaque occurence trouvée
dans la zone est affichée.

Exemple:
```
S DS:SI+10 L CX 'Hello',12,34
 Cette commande recherche la chaîne 'Hello' suivie
 par les octets 12H et 34H en commençant a l'offset
 SI+10 dans le segment de données actuel et en
 terminant CX octets plus loin.
```

F

F -- Remplir la mémoire avec des données

Syntaxe:

```
F adresse L longueur liste-données
liste-données -- liste d'octets ou chaînes séparées par
                   des virgules ou des espaces et encadrées
                   par des guillemets. Une chaîne peut
```

commencer avec une apostrophe seule ou
un guillemet.

longueur -- longueur en octets

Commentairse:
La commande F remplit la mémoire avec la série d'octets ou les caractères
précisés dans liste-données. La mémoire est remplie en commençant à l'adresse
précisée et pour la longueur précisée, en répétant liste-données si nécessaire.

Exemple:

```
F 8000:0 L 100 'Test'
Cette commande remplit la mémoire commençant à 8000:0 par
une longueur de 100H octets avec la chaîne 'Test'. La
chaîne 'Test' est répétée jusqu'à ce que la longueur à remplir
soit épuisée.
```

M

M -- Déplacement de données

Syntaxe:

```
M adresse-début L longueur adresse-fin
longueur -- longueur en octets
```

Commentaires:
La commande M déplace le nombre précisé d'octets de adresse-début dans la
mémoire vers adresse-fin dans la mémoire.

Exemple:

```
M 1000:0 L 200 2000:0
Cet ordre déplace 200H octets de l'emplacement
mémoire 1000:0 à l'emplacement mémoire 2000:0.
```

C

C -- Comparer deux blocs de données

Syntaxe:

```
C address1 L longueur address2
longueur -- longueur en octets
```

Commentaires:
La commande C compare le bloc mémoire spécifié par address1 et longueur
avec le bloc mémoire spécifié par address2 et longueur. Quand un octet du
premièr bloc de données est différend de son homologue dans le deuxième
bloc, les deux octets sont affichés avec leurs adresses.

Exemple:

```
C 5000:100 L 10 6000:100
```

Cette commande compare les 10H octets commençant à l'emplacement mémoire
5000:100 avec les 10H octets commençant à l'emplacement mémoire 6000:100.

5.6 Commandes Spécialisées de Débugging

```
Commandes:
 SHOW   -- instructions d'affichage du buffer d'historique
 TRACE  -- Entre en mode simulation de trace
 XT     -- Pas seul en mode simulation de trace
 XP     -- Pas de programme en mode simulation de trace
 XG     -- Va à l'adresse en mode simulation de trace
 XRSET  -- Reset du buffer de trace arrière
 VECS   -- Sauve/restaure/compare des vecteurs d'interruption
 SNAP   -- Fait une copie d'un bloc mémoire
 EMMMAP -- Affichage de la carte d'affectation EMM
```

SHOW

SHOW -- Affiche les instructions du buffer d'historique

Syntaxe:

```
   SHOW [B | début]
    B -- Cela indique à la commande SHOW de
              commencer l'affichage avec
              l'instruction la plus ancienne
              dans le buffer de trace arrière.
    début -- Le nombre d'instructions en arrière
              depuis la fin du buffer (la dernière
              instruction capturée) pour commencer
              l'affichage.
```

Commentaires:
La commande SHOW affiche les instructions du buffer historique de trace
arrière. Si le source est disponible pour les instructions alors l'affichage
est en mode mixte, autrement le code seul est affiché.

SHOW permet le défilement dans le buffer avec les touches haut, bas, Pageup
et PageDn. Pour sortir de SHOW vous devez presser la touche ESC.

Devant l'adresse de chaque instruction il y a le numéro d'entrée dans le
buffer. Ce nombre montre la profondeur dans le buffer que vous affichez.
Plus le nombre est grand, plus vous êtes profond dans le buffer.

Notes:
Avant d'utiliser la commande SHOW, les instructions doivent avoir été
enregistrées avec un domaine de trace arrière. Voir le chapitre 9 pour
plus d'information sur les domaines de race arrière.

Astuces:
Il est souvent utile d'avoir la fenêtre de code visible avec le code actuel
de la zone que vous affichez depuis le buffer de trace arrière. Quand vous
comparez le flux actuel d'instructions au code, les sauts et appels affichés
sont souvent moins déroutants.

En utilisant SHOW en conjonction avec la commande TRACE vous pourrez

voir les
instructions dans le buffer d'historique de trace de deux points de vue
différents.

Exemple:
  SHOW 40
   Cet exemple affiche en commençant avec la
   40ème instruction en arrière dans le buffer
   de trace arrière.

TRACE

TRACE -- Entre en mode simulation de trace

Syntaxe:

```
TRACE [début] | [OFF]
début -- Le nombre d'instructions a remonter
            depuis la fin du buffer (dernière
            instruction capturée) pour commencer
            une simulation de trace.
OFF -- Sortir du mode simulation de trace.
```

Commentaires:
La commande TRACE vous permet de 'rejouer' des instructions du buffer de trace
arrière tout comme si elles étaient exécutées pour la première fois. Pour
utiliser le mode simulation de trace vous devez avoir la fenêtre de code
visible. Après être entré dans le mode simulation de trace vous employez les
commandes XT, XP et XG pour tracer les instructions dans le buffer.

Pour sortir du mode simulation de trace entrez TRACE OFF.

TRACE sans paramètre affiche si le mode simulation de trace est ON ou OFF.

Note:
Avant d'utiliser la commande TRACE, les instructions doivent avoir été notées
avec un dommaine arrière de trace. Voir le chapitre 9 pour plus d'information
sur les domaines arrières de trace.

Astuce:
Le mode simulation de trace est plus utile quand la fenêtre de code est
visible. Il est souvent utile d'employer TRACE en conjonction avec la commande
SHOW. Cela permet de voir les instructions du buffer historique de trace de
deux façons différentes.

Exemple:
   TRACE 40
   Cet exemple entre en mode simulation de trace
   en commençant 40 instructions avant la dernière
   instruction notée. Il restera en mode simulation
   de trace jusqu'à ce que TRACE OFF soit entré.

XT

XT -- Saut d'une instruction en mode simulation de trace

Syntaxe:

    XT [R]
    R -- Saut d'une instruction en sens inverse

Commentaires:
La commande XT exécute pas à pas dans le buffer de trace arrière. Cette
commande se comporte comme la commande T pour le débugging normal.
Notez que les registres NE changent pas quand vous avancez en mode simulation
de trace sauf CS et IP.

L'instruction XT vous permet de 'rejouer' des instructions du buffer de trace
arrière.

Note:
Avant d'utiliser XT vous devez être dans en mode simulation de trace. Voir le
chapitre 9 et la commande TRACE dans cette section pour plus d'information sur
les domaines de trace arrière.

Astuce:
Si vous utilisez XT fréquemment, comme toute autre commande Soft-ICE, elle peut
être affectée à une touche de fonction.

Exemple:
    XT
    Cette commande exécute une seule instruction en mode simulation de trace.

XP

XP -- saut dans le programme en mode simulation de trace

La syntaxe:

    XP

Commentaires:
La commande XP fait un pas logique dans le programme dans le buffer de trace
arrière. Cette commande agit comme la commande P pour le débugging normal.
Notez que les registres NE changent pas en exécutant les instructions sauf CS
et IP.

L'instruction XP vous permet de 'rejouer' des instructions du buffer de trace
arrière.

Note:
Avant d'utiliser XP vous devez être en mode simulation de trace. Voir le
chapitre 9 et la commande TRACE dans cette section pour plus d'information sur
les domaines de trace arrière.

Astuce:
Si vous utilisez XP fréquemment, comme toute autre commande Soft-ICE elle peut
être affectée à une touche de fonction.

Exemple:
XP
Cette commande exécute un pas de programme en mode simulation de trace.

XG

XG -- Va à une adresse en mode simulation de trace

Syntaxe:

XG [R] adresse
R -- Recherche l'adresse en direction inverse

adresse -- Adresse à atteindre à dans le buffer de trace arrière.

Commentaires:
La commande XG déplace le pointeur d'instruction à l'occurence suivante
de l'adresse précisée dans le buffer de trace arrière. Si R est précisé
avant l'adresse, alors le pointeur d'instruction est déplacé à l'occurence
précédant l'adresse précisée dans le buffer de trace arrière.

L'adresse doit être le premier octet d'une instruction.

XG est analogue à la commande G dans un débugging normal.

Note:
Avant d'utiliser XG vous devez être en mode simulation de trace. Voir le
chapitre 9 et la commande TRACE dans cette section pour plus d'information
sur les domaines de trace arrière.

Exemple:
XG 273:1030
Cette commande déplace le pointeur d'instruction
à la prochaine instance de l'instruction à
l'adresse 273:1030.

XRSET

XRSET -- Remet à zéro le buffer de trace arrière

Syntaxe:

XRSET

Commentaires:
La commande XRSET remet a zéro le buffer de trace arrière. Cette commande
doit être exécutée avant de fixer un domaine de trace arrière s'il y a des
l'informations superflues dans le buffer.

Exemple:

XRSET
Cette commande remet le buffer de trace arrière à zéro.

VECS

VECS -- Sauve/Restaure/Compare des vecteurs d'interruption

Syntaxe:

```
VECS [C|S|R]
     C -- Comparer la table actuelle avec la table sauvegardée
     S -- Sauve la table d'interruption courante dans le buffer
     R -- Restaure la table d'interruption depuis le buffer
```

Commentaires:
La commande VECS vous permet de sauver et de restaurer la table d'interruptions dans un buffer interne de Soft-ICE. La table réelle peut aussi être comparée avec la table sauvée et les différences affichées.

Quand l'option C est employée pour comparer la table des vecteurs d'interruption courante avec la copie emmagasinée, la sortie est au format suivant:

adresse vieux-vecteur nouveau-vecteur

Tout vecteur ayant changé est affiché.

La table des vecteurs d'interruption est tout d'abord sauvée quand Soft-ICE est chargé. Elle est aussi sauvée automatiquement quand un programme est chargé avec LDR.EXE. Une seule copie de la table des vecteurs d'interruption est sauvée, donc chaque fois que VECS S est appelé, la copie précédente de la table d'interruption est effacée.

Si aucun paramètre n'est précisé, l'intégralité de la table des vecteurs d'interruption est affichée.

Exemple:
VECS C
Cette commande compare la table des vecteurs d'interruption actuelle avec une table qui avait été sauvegardée auparavant dans le buffer interne de VECS de Soft-ICE.

SNAP

SNAP -- Fait une copie d'un bloc de mémoire

Syntaxe:

```
SNAP [C | S | R] address1 address2
     C -- Comparer le buffer avec un domaine d'adresses
     S -- Sauver un domaine d'adresses dans le buffer
     R -- Restaurer le buffer vers un domaine d'adresses
```

Commente:
La commande SNAP fait une copie d'un bloc mémoire pour une comparaison

ultérieure. L'option S copie un bloc de mémoire vers un buffer dans la
mémoire étendue. L'option C affiche les différences entre le buffer dans
la mémoire étendue et la mémoire réelle spécifiée par le domaine d'adresses.
L'option R copie le buffer en mémoire étendue dans le domaine d'adresses
en mémoire conventionnelle.

Quand l'option C est employée pour comparer le buffer avec le domaine d'adresses la sortie est dans le format suivant:

adresse vieilles-données nouvelles-données

Chaque octet ayant changé est affiché.

L'adresse n'est normalement pas nécessaire pour les options C et R. Si
l'adresse n'est pas précisée, l'adresse du dernier SNAP qui a été entré
avec une adresse précisée sera employée.

Note:
Pour employer la commande SNAP, vous devez avoir précisé le commutateur
/TRA XXXX sur la ligne S-ICE.EXE dans CONFIG.SYS.

L'ordre SNAP sauve les données dans le buffer de trace arrière. Si vous
utilisez la trace arrière alors vous aurez un conflit avec SNAP.
Spécifiquement, SNAP efface les informations de trace arrière si vous faites
un SNAP S quand il y a des instructions dans le buffer de trace arrière.
Inversement, si vous avez sauvé une zone avec SNAP, l'activation d'un domaine
de trace arrière efface le buffer de SNAP.

Exemple:
SNAP S 2000:0 4000:0
Cette commande sauve le bloc de données
de 2000:0 à 4000:0 dans le buffer de
trace arrière de Soft-ICE.

EMMMAP

EMMMAP -- Affichage de la carte d'affectation EMM

Syntaxe:

EMMMAP

Commentaires:

La commande EMMMAP affiche chaque page physique disponible pour la mémoire
EMM et les pages qui sont actuellement mappées.

Note:
La caractéristique EMM de Soft-ICE doit être activée pour employer cette
fonction. Voir le chapitre 8 pour plus d'information sur l'activation

de
l'aptitude EMM.

Exemple:
  EMMMAP
   Cet exemple affiche l'allocation EMM courante dans la forme suivante.

```
   Phy page Seg adress Handle/Page
   00         D000      FFFF
   01         D400      0001/0000
   02         D800      0001/0001
   03         DC00      0001/0002
```

Dans cet exemple, la page physique 0 est située à D000 et est n'est pas mappée. La page physique 1 est située à D400 et a le handle 1, page 0 mappé sur elle. La page physique 2 est située à D800 et a le handle 1, page 1 mappé sur elle. La page physique 3 est située à DC00 et a le handle 2, page 1 mappé sur elle.

5.7 Commandes de fenêtrage

Commandes:
- WR -- Bascule la fenêtre registre
- WC -- Bascule/règle la taille de la fenêtre code
- WD -- Bascule/règle la taille de la fenêtre données
- EC -- Entre/sors de la fenêtre code
- . -- Trouve l'instruction courante

Trois types de fenêtres peuvent être créés avec Soft-ICE: registre, données, et code. Chacune de ces fenêtres peut être ouverte ou fermée à tout moment. Les fenêtres code et données peuvent être de taille variable; la fenêtre registre est de taille fixe. Les fenêtres restent toujours dans un ordre fixé. En partant du sommet de l'écran, l'ordre est: fenêtre registre, fenêtre données, puis fenêtre code.

WR

WR -- Bascule la fenêtre registre

Syntaxe:

 WR

Commentaires:
La commande rend la fenêtre registre visible si elle est actuellement invisible. Si la fenêtre registre est actuellement visible, WR l'efface.

La fenêtre registre affiche le jeu de registres du 8086 et les drapeaux du processeur.

Touche de fonction par défaut: F2

WC

WC -- Bascule/règle la taille de la fenêtre code

Syntaxe:

WC [taille-fenêtre]
taille-fenêtre -- un nombre décimal entre 1 et 21.

Commentaires:
Si taille-fenêtre n'est pas précisée, cette commande bascule la fenêtre code.
Si elle n'était pas visible elle est rendue visible, et si elle était visible
elle est effacée.

Si taille-fenêtre est précisée la taille de la fenêtre code est modifiée, ou
si elle n'était pas visible elle est rendue visible avec la taille précisée.

Note:
Si vous désirez déplacer le curseur dans la fenêtre code employez la commande EC. Voir la description de la commande EC pour plus de détails.

Exemple:
WC 12
Si aucune fenêtre code n'est présente,
alors une fenêtre code de 12 lignes de
hauteur est créée. Si la fenêtre code
est actuellement à l'écran, sa taille est
ajustée à 12 lignes.

WD

WD -- Bascule/règle la taille de la fenêtre données

Syntaxe:

WD [taille-fenêtre]
taille-fenêtre -- un nombre décimal entre 1 et 21.

Commentaires:
Si taille-fenêtre n'est pas précisée, cette commande bascule la fenêtre données.
Si elle n'était pas visible elle est rendue visible, si elle était visible elle
est effacée.

Si taille-fenêtre est précisée la taille de la fenêtre données est modifiée,
ou si elle n'était pas visible elle est rendue visible avec la taille précisée.

Exemple:
WD 1
Si aucune fenêtre données n'est présente,
alors une fenêtre données de 12 lignes de
hauteur est créée. Si la fenêtre données
est actuellement à l'écran, sa taille est
ajustée à 1 ligne.

EC

EC -- Entre/sors de la fenêtre code

Syntaxe:

```
    EC
```

Commentaires:
La commande EC bascule le curseur entre la fenêtre code et la fenêtre de commandes. Si le curseur était dans la fenêtre de commandes il est déplacé dans la fenêtre code, et si le curseur était dans la fenêtre code il est déplacé dans la fenêtre de commandes.

Quand le curseur est dans la fenêtre code plusieurs options deviennent disponibles et cela rend le débugging beaucoup plus facile.
Les options sont:

* Le 'pointer-et-activer' les point d'arrêt
  Le 'pointer-et-activer' les points d'arrêt est réalisé avec la commande BP. Si aucun paramètre n'est précisé avec la commande BPX un point d'arrêt d'exécution est positionné à la position du curseur dans la fenêtre code. Le curseur doit être sur une ligne qui contient du code (placer la fenêtre de code en mode mixte si vous n'êtes pas sûr).
  L'affectation par défaut pour BPX est F9.

* Aller à la ligne du curseur
  Vous pouvez établir un point d'arrêt temporaire au curseur et y aller avec la commande HERE. Le curseur doit être sur une ligne qui contient du code (placer la fenêtre code en mode mixte si vous n'êtes pas sûr).
  L'affectation par défaut pour HERE est F7.

* Défilement de la fenêtre code
  La fenêtre code ne peut être scrollée que si le cursor est à l'intérieur. Les touches de défilement (flèche haut, flèche bas, PageUp et PageDown) sont redéfinies quand le curseur est dans la fenêtre code. Quand le curseur est dans la fenêtre code les touches de défilement se comportent ainsi:

```
  haut   -- la fenêtre code scrolle vers le haut d'une ligne
  bas    -- la fenêtre code scrolle vers le bas d'une ligne
  pageup -- la fenêtre code scrolle vers le haut d'une fenêtre
  pageDn -- la fenêtre code scrolle vers le bas d'une fenêtre
```

Note:
La fenêtre code doit être visible pour la commande EC fonctionne. La touche de fonction par défaut: F6

.

. -- Trouver l'instruction courante

Syntaxe:
 .

Commentaires:
Quand la fenêtre code est visible, la commande . affiche la ligne actuelle
du source ou l'instruction actuelle.

5.8 Commandes de personnalisation du Débugger

```
Commandes:
   PAUSE  -- Pause après chaque écran
   ALTKEY -- Modifie la séquence de touches servant à
                  activer Soft-ICE
   FKEY   -- Affiche et édite les touches de fonction
   BASE   -- Positionne/affiche la base courante de numération
   CTRL-P -- Bascule la session vers l'imprimante
Print-Screen -- Imprime le contenu de l'écran
   PRN    -- Positionne le port d'imprimante
```

PAUSE

PAUSE -- Pause après chaque écran

Syntaxe:

PAUSE [ON | OFF]

Commentaires:
PAUSE contrôle les pauses écran au bout de chaque page. Si PAUSE est ON,
on vous propose de presser une touche avant l'affichage de la page suivante.
Le prompt est affiché dans la ligne de statut en bas de la fenêtre.

Si aucun paramètre n'est précisé, l'état actuel de PAUSE est affiché.

Le mode par défaut de PAUSE est ON.

Exemple:
   PAUSE ON
   Cette commande précise que les commandes
   d'affichage dans les fenêtres provoqueront
   une attente touche après l'affichage de
   chaque écran.

ALTKEY

ALTKEY -- Modifie la séquence de touches servant à activer Soft-ICE

Syntaxe:

```
     ALTKEY [ALTlettre] | [CTRLlettre] | [SYSREQ]
     lettre -- Toute lettre (A - Z)
```

Commentaires:
La commande ALTKEY permet de modifier la séquence de touches pour activer
Soft-ICE. La séquence de touches peut être changée en CTRL + lettre,
ALT + lettre, ou la touche SysRq.

Occasionnellement vous pouvez utiliser un programme qui entre en conflit
avec la séquence de touches CTRL D qui active Soft-ICE. Une méthode pour

résoudre ce problème est d'utiliser la commande ALTKEY pour changer la
séquence de touches. Une autre méthode est d'ajouter la touche SHIFT à la
séquence actuelle. Soft-ICE ne répond pas à cette séquence de touches et
permet d'accéder à votre programme. Par exemple si un programme résident que
vous utilisez est activé avec la séquence de touches CTRL D, essayer d'utiliser
la séquence de touches CTRL SHIFT D pour activer votre programme résident. Sur
certains claviers, vous devez presser ALT et la touche PrtScr simultanément
pour générer un Sys Request.

Si aucun paramètre n'est précisé, la séquence de touches actuelle est affichée.

La séquence de touches par défaut est CTRL D.

Exemple:

ALTKEY ALT Z

Cette commande précise que la séquence de touches ALT Z sera maintenant utilisée pour activer Soft-ICE

FKEY

FKEY -- Affichage et édition des touches de fonction

Syntaxe:

```
FKEY [nom-touche-fonction chaîne]
nom-touch-fonction -- F1, F2... F12

chaîne -- La chaîne consiste en toute commande valable
                  de Soft-ICE et les caractères spéciaux
                  ^ (circonflexe) et ; (point-virgule).
                  Un ^ est placé dans la chaîne pour faire
                  une commande invisible. Un ; est placé
                  dans la chaîne pour simuler un retour de
                  chariot.
```

Commentaires:

La commande FKEY est employé sur la ligne de commandes pour affecter une
touche de fonction à une chaîne de commande. La touche de fonction peut être
affectée à toute chaîne de commande pouvant être entrée dans Soft-ICE.

Si aucun paramètre n'est précisé, alors les affectations actuelles des touches
de fonction sont affichées.

Pour désafecter une touche de fonction particulière, utilisez la commande
FKEY avec ces paramètres: un nom de touche de fonction suivi par une chaîne
nulle.

Les touches de fonction peuvent aussi être pré-initialisées dans le fichier de
définition S-ICE.DAT. Pour plus d'informations sur les définitions de

touches
de fonction dans le fichier de définition, référez-vous à la section 6.4.

L'utilisation du symbole de retour chariot dans les chaînes d'affectation
des touches de fonction vous permet d'attribuer à une touche de fonction
une série de commandes. Un retour chariot est représenté par un ; (point-virgule).

Si vous mettez ^ (Shift 6) au début d'une définition de touche de fonction,
la commande correspondante sera invisible. La commande fonctionnera
normalement, mais toute information affichée dans la fenêtre de commande
(messages d'erreur inclus) est supprimée. Le mode invisible est utile quand
une commande change une information dans une fenêtre (code, registre ou
données) mais que vous ne voulez pas encombrer la fenêtre de commandes.

Quand une touche de fonction est rendue invisible avec ^, la touche de
fonction peut être employée au milieu de l'entrée d'une autre commande
sans affecter cette entrée. Par exemple, si vous utilisez l'affectation
par défaut pour F2, vous pouvez basculer la fenêtre de registre avec F2
même si vous êtes en train d'entrer votre prochaine commande.

Note:
Soft-ICE a maintenant un fichier de définition nommé S-ICE.DAT. Vous pouvez
placer des affectations de touches de fonction dans ce fichier si bien
que ces touches de fonction seront automatiquement affectées quand
Soft-ICE sera chargé. La syntaxe pour affecter une touche de fonction dans
le fichier de configuration est:

```
 nom-touche-fonction = "chaîne"
```

Quand vous affectez une touche de fonction à une chaîne de commande dans
S-ICE.DAT, la chaîne doit être incluse dans des guillemets.

Exemple de ligne de commande:

```
   FKEY F2 ^WR;
    Cet exemple attribuera la commande de
    basculage de la fenêtre registre a
    la touche F2. Le ^ rend la commande
    invisible, et le ; termine la commande
    avec un retour de chariot. La touche
    F2 bascule la fenêtre registre ON ou
    OFF, et peut même être activée pendant
    l'entrée d'une autre commanden.

   FKEY F1 "G CS:120; R; G CS:"
    Cet exemple montre que des commandes
    multiples peuvent être attribués à une
    seule touche de fonction et que des
```

ordres partiels peuvent être affectés
pour être complétés par l'utilisateur.
L'entrée de cette commande, en pressant
la touche F1 forcera le programme à
s'exécuter jusqu'à ce que l'emplacement
CS:120 soit atteint, affichera les
registres, commencera alors la commande
G pour qu'lle soit complétée par
l'utilisateur.

FKEY F1 WD 3; D DS:100;
Cet exemple attribuera une série de
commandes à la touche F1. La fonction est
visible, et finit avec un retour de chariot.
La touche F1 positionnera la fenêtre données
à trois lignes de haut et fera un dump des
données en commençant à DS:100.

S-ICE.DAT exemple:
F1 = "WR;WD 2;WC 10;"
Si cette ligne est placée dans S-ICE.DAT, quand
Soft-ICE est chargé il affecte la chaîne à la
touche F1. Quand F1 est pressée dans Soft-ICE,
elle bascule la fenêtre registre, crée une
fenêtre données de hauteur 2 et une fenêtre
code de hauteur 10. Pour plus d'information sur
l'affectation des touches de fonction dans
S-ICE.DAT, se référer au chapitre 6.

BASE

BASE -- Positionne la base de numération courante

Syntaxe:

BASE [10 | 16]

Commentaires: La commande BASE fixe la base courante de numération à base 10
ou base 16. La base 10 est d'emploi limité dans la fenêtre étroite à cause
des limitations de largeur de la fenêtre. Elle limite aussi la quantité
d'informations affichées par certaines commandes en mode large.

Quand la base courante est base 10, tous les nombres et les adresses
entrés et affichés par Soft-ICE sont en décimal. Quand la base courante est
base 16, tous les nombres et adresses entrés et affichés par Soft-ICE sont
en hexadécimal sauf:

* les numéros des lignes de source
* les coordonnées écran et les tailles dans la commande WIN

Ces exceptions sont toujours entréees et affichées comme nombres
décimaux.

La base par défaut est base 16.

Exemple:
BASE 16
Cet exemple positionne la base courante à base 16.

CTRL - P

CTRL-P -- Bascule l'enregistrement de la scéance sur l'imprimante

Syntaxe:

  CTRL-P

Commentaires:
Quand la touche CTRL suivie par la touche P est pressée, toute information
affichée dans la fenêtre de commande est aussitôt envoyée à l'imprimante.
Pour basculer l'impression OFF, entrez CTRL suivi par P une deuxième fois.

Quand vous envoyez des informations à l'imprimante en utilisant CTRL-P,
vous pouvez vouloir basculer la commande PAUSE sur OFF pour permettre à
l'information de défiler dans la fenêtre sans presser une touche.

Print-Screen

Print-Screen - Imprime le contenu de l'écran

Syntaxe:

   Print-Screen

Commentaires:
L'appui sur la touche Print-Screen fait une copie d'écran sur l'imprimante.
Toutes les informations de l'écran sont envoyées à l'imprimante.

Si vous désirez imprimer la carte de la mémoire ou des information d'aide,
il est habituellement beaucoup plus rapide à d'employer CTRL-P que Print-Screen.
C'est parce Print-Screen imprime tous les caractères sur l'écran y compris les
bordures.

PRN

PRN --- Positionne le port de sortie imprimante

Syntaxe:

```
PRN [LPTx | COMx]
x -- un nombre décimal entre 1 et 4.
```

Commentaires:
La commande PRN vous permet envoyer la sortie de CTRL-P et de Print-Screen
à un port imprimante différent.

Si aucun paramètre n'est entré, PRN affiche le port d'imprimante actuellement attribué.

Exemple:

```
PRN COM 1
 Cette commande envoie les sorties de
 CTRL-P et Print-Screen vers le port COM 1.
```

5.9 Commandes de Contrôle D'écran

```
Commande:
 FLASH         -- Restaure l'écran durant P et T
 FLICK         -- Réduction du 'Flicker'
 WATCHV        -- Positionne le mode de surveillance vidéo
 RS            -- Restaure l'écran de programme
 CLS           -- Efface la fenêtre
 ALTSCR        -- Changement d'écran
 WIN           -- Changement de taille de la fenêtre Soft-ICE
```

FLASH

FLASH -- Restaure l'écran durant P et T

Syntaxe:

FLASH [ON | OFF]

Commentaires: L'ordre FLASH vous permet de précisez comment l'écran sera
restauré durant les commandes Saut de Programme et Trace. Si vous précisez que
l'écran doit être restauré, il est restauré durant la brève période de temps
que met la commande P ou T à s'exécuter. Cette caractéristique est requise pour
débugger des sections de code qui accèdent à la mémoire vidéo.

Si la commande P exécute le contenu d'un saut ou d'une interruption, l'écran
sera toujours restauré, parce que la routine étant appelée peut écrire sur
l'écran.

Si aucun paramètre n'est précisé, l'état actuel de FLASH est affiché.

Le mode par défaut de FLASH est OFF.

Exemple:
  FLASH ON
   Cette commande met ON le mode FLASH. L'écran
   sera restauré durant toute commande P ou T.

FLICK

FLICK -- réduction du scintillement de l'écran

Syntaxe:

FLICK [ON | OFF]

Commentaires:
Certains types de cartes vidéo requièrent une attente des rafraîchissements
horizontaux ou verticaux avant d'afficher des caractères. Si les écritures
vidéo sont faites arbitrairement, un scintillement apparaîtra pendant
l'affichage des caractères. Si un scintillement survient sur votre écran
lors de l'utilisation de Soft-ICE, vous devriez positionner FLICK sur ON.

Avec certaines cartes EGA, les couleurs ne sont pas restaurées correctement
quand vous sortez de Soft-ICE. C'est un problème de virtualisation de la
vidéo EGA. Le port 3DA est un port vidéo employé dans deux buts. Le premier
est les vieux softs CGA testant 3DA pour hsync et vsync. Cela leur permet
de ne pas avoir de scintillement avec quelques vieilles cartes contrôleur CGA.
Le second est qu'il est employé à la remise à zéro d'un verrou de palette
sur les cartes EGA. Soft-ICE a un algorithme qui lui permet d'éviter d'avoir
à constamment surveiller ce port, ce qui pourrait ralentir de vieux programmes
qui penseraient qu'ils sont en CGA. Cependant, il peut occasionnellement y
avoir des circonstances où cet algorithme ne fonctionne pas. Si vous utilisez
Soft-ICE sur un écran EGA et si vous remarquez que les couleurs ne sont pas
restaurées correctement, alors positionnez FLICK sur ON et Soft-ICE surveillera
le port 3DA, fixant le problème.

Quand FLICK est sur ON, les mises à jour d'écran sont plus lentes.

Si aucun paramètre n'est précisé, l'état actuel de FLICK est affiché.

Le mode par défaut de FLICK est OFF.

Exemple:

FLICK ON

Cette commande le mode FLICK sur ON. Cela force
Soft-ICE à attendre pour les rafraîchissements
horizontal ou vertical avant d'afficher des caractères.

WATCHV

WATCHV -- Affiche le mode surveillance vidéo

Syntaxe:

WATCHV [ON | OFF]

Commentaires:
La commande WATCHV vous permet préciser comment Soft-ICE doit surveiller les
ports vidéo. Normalement, Soft-ICE ne surveille les ports vidéo qu'après
qu'une INT10 ait été exécutée si elle bascule en mode graphique. Certains
programmes n'emploient pas INT10 pour changer de mode. Dans ces cas-là, si
WATCHV est OFF, Soft-ICE peut avoir des problèmes pour sauver et restaurer
l'écran correctement. En basculant WATCHV ON on oblige Soft-ICE à surveiller
les ports vidéo tout le temps.

Basculez WATCHV sur ON si vous remarquez que Soft-ICE ne gère pas votre écran
correctement, ou si le curseur n'est pas restauré correctement. Le

fait de
basculer WATCHV sur ON peut avoir des effets sur les performances dans certains
modes vidéo.

Si aucun paramètre n'est précisé, l'état actuel de WATCHV est affiché.

Le mode par défaut de WATCHV est OFF.

Exemple:
WATCHV ON
Cette commande met WATCHV en mode ON. Cela force
Soft-ICE à surveiller des ports vidéos supplémentaires
dans un but de virtualisation.

RS

RS -- Restaure l'écran de programme

Syntaxe:

RS

Commentaires:
La commande RS vous permet de restaurer l'écran du programme temporairement.
La fenêtre Soft-ICE fenêtre disparaît jusqu'à ce qu'une touche soit pressée.
Cette caractéristique est utile en débuggant des programmes graphiques qui
mettent à jour l'écran fréquemment. Quand Soft-ICE est réactivé, il revient
en mode texte. L'utilisation de la commande RS restaure temporairement l'écran
graphique.

Exemple:
RS

CLS

CLS -- Effacement de la fenêtre

Syntaxe:

CLS

Commentaires:
La commande CLS efface la fenêtre Soft-ICE et déplace le prompt et le curseur
dans le coin supérieur gauche de la fenêtre.

Exemple:
CLS

ALTSCR

ALTSCR -- Changement d'écran

Syntaxe:

ALTSCR [ON | OFF]

Commentaires:
La commande ALTSCR vous permet de rediriger les sorties écran de Soft-ICE
de votre écran par défaut à l'écran alterné. Cette caractéristique est utile,
par exemple, quand vous voulez débugger un programme graphique sans devoir
changer entre la fenêtre Soft-ICE et l'affichage graphique.

ALTSCR nécessite que le système ait deux moniteurs connectés. Le moniteur
alterné doit être en mode caractère, qui est le mode par défaut pour les
moniteurs.

Le mode par défaut de ALTSCR est OFF.

Exemple:
   ALTSCR ON

Cette commande redirige les sorties écran vers le moniteur alterné.

WIN

WIN -- Change la taille de la fenêtre Soft-ICE

Syntaxe:

```
   WIN [N | W] [début-rangée longueur [début-colonne]]
   N -- Quand N est précisé, la fenêtre est mise à la
                largeur étroite: 46 caractères.
   W -- Quand W est précisé, la fenêtre est positionnée
                à la largeur plein écran.
   début-rangée -- Nombre de 0 à 17 précisant la rangée
                où débute l'affichage de la fenêtre.
   longueur -- Nombre de 8 à 25 précisant de combien
                de combien de lignes vous voulez la fenêtre.
   début-colonne -- position de la colonne gauche de la fenêtre
                étroite. Début-rangée et début-colonne précisent
                le coin supérieur gauche de la fenêtre
                étroite. Début-colonne est ignoré s'il
                est appliqué à la fenêtre large.
```

Commentaires:
La commande WIN vous permet modifier la largeur et la hauteur de la
fenêtre Soft-ICE.

Si aucun paramètre n'est précisé, cette commande bascule la fenêtre entre
l'affichage étroit et l'affichage large.

Si la commande WIN est précisé avec seulement le paramètre N ou W, la taille
de fenêtre sera changée à la largeur demandée et à la hauteur actuelle.

Si le nombre de lignes plus le numéro de la rangée de départ est plus grand
que 25, la fenêtre va jusqu'au bas de l'écran.

Le mode par défaut de WIN est NARROW.

Exemples:

```
WIN N 4 9 30
Cette commande force l'affichage à commencer
à la rangée 4 et à la colonne 30, et à être
haut de 9 lignes et large de 46 caractères.
WIN
Cette commande bascule la largeur de la fenêtre
de son état actuel (large ou étroit) à l'état
contraire.
WIN W 10 8
Cette commande force la fenêtre à commencer
à la colonne 10, et à être haut de 8 lignes
et de toute la largeur de l'écran.
```

5.10 Commandes symboles et ligne source

```
Commandes:
 SYM        -- Affiche/Positionne les symboles
 SYMLOC     -- Déplace la base de symboles
 SRC        -- Bascule entre source, mélangé et code
 FICHIER    -- Change/affiche le source actuel
 SS         -- Cherche une chaîne dans le fichier source actuel
```

158

SYM

SYM -- Affiche/positionne les symboles

Syntaxe:

```
SYM [nom-symbole [valeur]]
Nom-symbole -- Un nom de symbole valide. Le nom du symbole
                peut se terminer avec un * (astérisque).
                Cela permet de chercher un symbole si on
                ne connait que la première partie de son
                nom. Le caractère , (virgule) peut être
                employé comme un joker à la place d'un
                caractère dans nom-symbole.

valeur -- C'est une valeur de mot qui est employée si vous
                voulez positionner un symbole à une valeur
                spécifique.
```

Commentaires:
La commande SYM permet d'afficher et de positionner des symboles. Si SYM
est entré sans paramètre tous les symboles sont affichés. La valeur de chaque
symbole est affichée à côté du nom du symbole.

Si un nom de symbole est entré sans aucune valeur alors la valeur et nom du
symbole sont affichés. Si le nom du symbole n'est pas trouvé alors rien n'est
affiché.

La commande SYM est souvent utile pour trouver un nom de symbole quand vous
ne pouvez vous rappeler que d'une portion du nom. Deux types de jokers sont
disponibles pour situer des symboles. Si nom-symbole finit avec un *, alors
tous les symboles qui ont les mêmes caractères précédant le * seront affichés

indépendamment de leurs terminaisons. Si une virgule est employée à la place
d'un caractère spécifique dans nom-symbole, cette virgule est un joker.

Si valeur est précisée, tous les symboles correspondant à nom-symbole sont
positionnés à la valeur. Tous les symboles ont des valeurs de mot.

Exemples:
   SYM FOO *
   Tous les symboles commençant par FOO sont affichés.
   SYM FOO * 6000
   Tous symboles commençant par FOO sont positionnés à la valeur 6000.

SYMLOC

SYMLOC -- Reloge la base de symboles

Syntaxe:

SYMLOC segment-adresse

Commentaires:
La commande SYMLOC reloge l'adresse de segment de tous les symboles par rapport
à l'adresse de segment précisée. Cette fonction est nécessaire en débuggant
de pilotes de périphériques ou d'autres programmes qui ne peuvent pas être
chargés directement avec LDR.EXE.

Quand on reloge un pilote de périphérique, il faut employer la valeur de
l'adresse de base du pilote comme trouvé avec la commande MAP. Quand on reloge
un programme .EXE, la valeur est 10H plus élevée que celle trouvée comme base
avec la commande MAP. Quand on reloge un programme .COM, il faut employer
l'adresse de base du segment qui est trouvée la commande MAP.

La commande MAP affiche au moins deux entrées pour chaque programme. La première
est typiquement l'environnement et le second est typiquement le programme.
L'adresse de base du programme est la valeur de relogement.

Exemple:
    SYMLOC 1244 + 10
    Ceci reloge tous les segments de la table de symbole
    à 1244. Le + 10 est employé pour reloger un TSR qui
    était à l'origine un fichier .EXE. S'il s'était agi
    d'un fichier .COM le + 10 n'était pas nécessaire.

SRC

SRC -- Bascule entre source, mélangé et code

Syntaxe:

  SRC [?]

Commentaires:
La commande SRC bascule entre les modes source, mélangé et code dans la
fenêtre code.

Si SRC? est entré, l'état actuel est affiché.

Exemple:
```
SRC
Cette commande change la mode actuel de la
fenêtre de code. Si le mode était source,
il devient mixte. Si le mode était mixte,
il devient code. Si le mode était code,
il devient source.
```

La touche de fonction par défaut: F3

FILE

FILE -- Modifie/affiche le fichier source courant

Syntaxe:

FILE [nom-fichier]
Commentaires:
Si nom-fichier est précisé, ce fichier devient le fichier courant et le
début du fichier est affiché dans la fenêtre code. Si aucun nom n'est précisé,
le nom du fichier source actuel (s'il existe) est affiché.

La commande FILE est souvent utile quand on positionne un point d'arrêt
sur une ligne qui n'a aucun symbole public associé. Utilisez FILE pour
amener le fichier désiré dans la fenêtre code, employez la commande SS pour
situer la ligne spécifique, déplacer le curseur sur la ligne spécifique, puis
entrez alors BPX pour positionner le point d'arrêt.

Note:
Seuls les fichiers source chargés en la mémoire étendue avec LDR.EXE sont
disponibles pour la commande FILE.

Exemple:
```
FILE MAIN.C
Si MAIN.C avait été chargé avec LDR.EXE, cette
commande l'amène dans la fenêtre de code en
commençant par la ligne 1.
```

SS

SS -- Cherche une chaîne dans le fichier source actuel

Syntaxe:

SS [numéro-ligne] ['chaîne']

numéro-ligne -- un nombre décimal

chaîne -- une chaîne de caractère entourée
par des apostrophes ou des guillemets.

Commentaires:
La commande SS cherche la chaîne de caractères précisée dans le fichier source.
S'il y a une occurence, la ligne dans laquelle la chaîne est située est
affichée sur la première ligne dans la fenêtre code.

La recherche commence au numéro de ligne précisé. Si aucun numéro de ligne
n'est précisé la recherche débute à la première ligne affichée dans la fenêtre
code.

Si aucun paramètre n'est précisé, la recherche continue pour la précédente
chaîne.

Note:
La fenêtre code doit être visible et en mode source avant d'utiliser la
commande SS.

Exemple:

```
    SS 1 'if(i = = 3)'
    La chaîne 'if(i = = 3)' est recherchée dans le
    fichier source actuel en commençant à la ligne 1.
    La ligne contenant la prochaine occurence de la
    chaîne devient la première ligne affichée dans
    la fenêtre de code.
```

CHAPITRE 6

Options d'Initialization de Soft-ICE

6.1 Introduction

Le programme Soft-ICE (S-ICE.EXE) peut être chargé comme un pilote de périphérique dans CONFIG.SYS ou comme un programme depuis la ligne de commande DOS. Pour exploiter Soft-ICE au maximum, il doit être chargé initialement comme un pilote de périphérique dans CONFIG.SYS. Cependant, il peut y avoir des circonstances où vous pourriez vouloir lancer Soft-ICE depuis le prompt DOS ou dans un fichier .BAT:

* Vous n'avez pas de mémoire étendue dans votre système. Soft-ICE ne peut être chargé comme un pilote de périphérique que si vous avez de la mémoire étendue.
* Vous ne voulez utiliser AUCUN octet de mémoire conventionnelle. Quand il est chargé comme un pilote de périphérique, Soft-ICE occupe à-peu-près 2K de mémoire conventionnelle.
* Vous ne pensez employer Soft-ICE qu'occasionnellement et vous n'avez pas d'autres programmes utilisant la mémoire étendue.

Dans certains cas vous pouvez avoir besoin de certaines caractéristiques qui imposent à Soft-ICE d'être chargé dans CONFIG.SYS mais vous ne voulez pas que Soft-ICE soit résidant tout le temps. Dans ce cas Soft-ICE peut être chargé dans CONFIG.SYS pour réserver de la mémoire étendue, puis désactivé, en utilisant le commutateur /UN, jusqu'à ce que Soft-ICE soit nécessaire. Voir la section 6.3.1 pour plus d'informations sur le commutateur /UN.

6.2 Chargement de Soft-ICE depuis le prompt du DOS

Vous NE pouvez PAS activer toutes les caractéristiques de Soft-ICE en le chargeant depuis le prompt DOS. Si vous utilisez Soft-ICE comme un débugger seul, il est recommandé de le charger dans le fichier CONFIG.SYS.

Pour charger Soft-ICE depuis le prompt du DOS entrez:

S-ICE

Dans les systèmes sans mémoire étendue, Soft-ICE se charge lui-même dans le plus haut emplacement mémoire possible. La mémoire utilisée par Soft-ICE est alors rendue invisible aux programmes DOS. Malgrès que la mémoire totale visible par DOS et ses programmes soit moindre après le chargement de Soft-ICE, il est recommandé de charger Soft-ICE avant tout programme T&SR.

Dans des systèmes avec de la mémoire étendue, vous ne devriez charger Soft-ICE depuis le prompt DOS que si vous n'avez pas à utiliser la mémoire étendue pour autre chose (ex, VDISK, CACHE, HIMEM...). Quand vous chargez initialement Soft-ICE depuis la ligne de commande ou depuis un fichier .BAT, Soft-ICE vous avertira avec un message. Ce message d'avertissement ne sert qu'à vous rappeler que Soft-ICE écrase la zone la plus haute de la mémoire étendue quand il se charge. Vous pouvez supprimer cet avertissement avec l'option EXTENDED dans le fichier de configuration (S-ICE.DAT). Pour plus d'information sur l'option EXTENDED, voir la section 6.4.1.

6.3 Chargement de Soft-ICE comme un pilote de périphérique

Dans le but de rendre disponibles toutes les caractéristiques de Soft-ICE, vous devez d'abord charger Soft-ICE comme un pilote de périphérique dans votre fichier CONFIG.SYS. Les caractéristiques rendues disponibles sont:

* Coexister avec autre logiciel qui emploie la mémoire étendue.
  Le chargement comme un pilote de périphérique permet à Soft-ICE de gérer la mémoire étendue ainsi vous pouvez lancer Soft-ICE avec des programmes qui emploient la mémoire étendue, tels que VDISK, CACHE et HIMEM.
* Débugging Symbolique et niveau source
  Le chargement comme un pilote de périphérique permet à Soft-ICE d'affecter un buffer en mémoire étendue pour les informations de source et symboles.
* Les domaines de trace arrière et la commande SNAP

Le chargement comme un pilote de périphérique
permet à Soft-ICE d'affecter un buffer en mémoire
étendue pour un buffer de trace arrière. Ce buffer
est aussi utilisé pour la commande Soft-ICE SNAP.

* Activer les possibilités EMM 4.0 de Soft-ICE
* Faire tourner Soft-ICE avec MagicCV ou MagicCVW

Notes:
Quand il est chargé comme un pilote de périphérique dans CONFIG.SYS, Soft-ICE
affecte la zone la plus haute de mémoire étendue pour lui-même et ses composants associés, afin qu'il ne puisse y avoir aucun conflit de mémoire.
Le fichier S-ICE.EXE doit être chargé dans CONFIG.SYS avant tout autre
pilote qui affecte la mémoire étendue (ex, VDISK.SYS, RAMDRIVE.SYS).
Généralement Soft-ICE travaille mieux s'il est le premier pilote de périphérique installé dans CONFIG.SYS.

6.3.1 Commutateurs de chargement de Soft-ICE

Un ou plusieurs commutateurs de chargement peuvent suivre S-ICE.EXE dans
CONFIG.SYS. Ces commutateurs vous permettent personnaliser la façon dont la
mémoire étendue sera réservée par Soft-ICE. Tous les commutateurs doivent
commencer par le caractère /.
Les commutateurs de chargement sont:

* /EXT XXXX -- Demande à S-ICE.EXE de réserver XXXX kilo-octets de mémoire étendue pour d'autres programmes DOS qui utilsent la mémoire étendue (ex, VDISK, CACHE, HIMEM,...). Si le commutateur /EXT n'est pas présent, alors toute la mémoire étendue que n'emploient pas Soft-ICE ou ses composants associés sera laissée comme mémoire étendue standard, mais la quantité ne peut pas être garantie. Le commutateur /EXT est utile parce qu'il est parfois difficile de déterminer exactement combien de mémoire est utilisée par Soft-ICE et ses composants associés. L'utilisation du commutateur /EXT garantira que la quantité précisée sera disponible pour d'autres programmes utilisant la mémoire étendue.

* / SYM XXXX -- Demande à S-ICE.EXE de réserver XXXX kilo-octets de mémoire étendue pour le source et les symboles. Si XXXX n'est pas précisé, alors toute la mémoire étendue restante est utilisée pour les symboles. Assez de mémoire doit être affectée pour votre fichier .SYM et tous les fichiers source. Pour plus d'informations sur l'utilisation des sources et des symboles, voir le chapitre 7.

* /TRA XXXX -- Demande à S-ICE.EXE de réserver XXXX kilo-octets de mémoire étendue pour un buffer d'historique de trace. Ce buffer est utilisé pour des domaines de trace arrière et pour la commande SNAP. Si XXXX n'est pas précisé, alors 10K de mémoire étendue sont automatiquement réservés pour le buffer. Si vous ne voulez pas du tout de mémoire réservée

pour un buffer de trace arrière, utilisez /TRA 0.
Pour plus d'informations sur l'utilisation des domaines de trace arrière, voir le chapitre 9.

* /MCV XXX -- Demande à S-ICE.EXE de réserver XXX kilo-octets de mémoire étendue pour MagicCV ou MagicCVW. La quantité minimum de mémoire étendue que vous pouvez affecter est 280K et le maximum est 620K. Si XXX n'est pas précisé, S-ICE.EXE réservera la mémoire restante, entre 280K et 620K. Voir le chapitre 10 pour plus d'informations sur le lancement de Soft-ICE avec MagicCV ou MagicCVW.

* /EMM XXXX -- Demande à S-ICE.EXE de transformer XXXX kilo-octets de mémoire étendue en mémoire paginée EMM 4.0. Si XXXX est précisé, alors toute la mémoire restante est uttilisée comme paginée. Voir le chapitre 8 pour plus d'informations sur le support de la mémoire paginée.

* /UN -- Demande à S-ICE.EXE d'entrer en mode protégé, de réserver toute la mémoire étendue nécessaire, de sortir du mode protégé et de se décharger lui-même. Ce commutateur doit être utilisé quand vous chargez S-ICE.EXE comme un pilote de périphérique, mais vous ne voulez pas que votre système reste en mode protégé. Ce commutateur réserve la mémoire pour Soft-ICE, et vous devrez exécuter S-ICE.EXE au prompt du DOS quand vous êtes prêts à employer Soft-ICE.

Soft-ICE réserve la mémoire étendue dans l'ordre suivant, indépendamment de
l'ordre dans lequel les commutateurs sont précisés:

Réserve à-peu-près 120K pour S-ICE.EXE.
Réserve de la mémoire pour le commutateur /EXT si présent.
Réserve de la mémoire pour le commutateur /SYM si présent.
Réserve de la mémoire pour le commutateur /TRA si présent.
s'il n'est pas présent, réserve par défaut 10K pour le buffer de trace arrière.
Réserve de la mémoire pour le commutateur /MCV si présent.
Réserve de la mémoire pour le commutateur /EMM si présent.

Si la mémoire disponible venait à manquer tandis que l'on essaye de réserver de la mémoire pour un commutateur dans la séquence au-dessus, alors S-ICE.EXE procède ainsi:

1. La reste de la mémoire étendue est affecté pour switcher quand la mémoire manquera.
2. Aucune mémoire ne sera réservée pour les commutateurs restants.

Notes:

Si le commutateur /MCV ou /EMM est présent, 64k de mémoire étendue supplémentaire sont réservés pour un buffer DMA.

Les commutateurs peuvent être placés dans nimporte quel ordre après la
commande DEVICE = S-ICE.EXE. Un exemple:

```
DEVICE = S-ICE.EXE /TRA50 /EMM 500 /SYM 2048
```

Si 4 méga-octets de mémoire étendue sont disponibles, cet exemple

réservera à peu près 120K pour Soft-ICE, 2 méga-octets pour des symboles,
50K pour un buffer de trace arrière, 500K pour la mémoire paginée et laisse
à peu près 1.3 méga-octets pour autres programmes utilisant la mémoire
étendue. Notez que Soft-ICE se chargera dans la zone la plus haute de mémoire
étendue, en laissant le reste la mémoire commençant à 100000H (un méga-octet).

6.4 Le fichier d'initialisation S-ICE.DAT

Soft-ICE a plusieurs options de chargement. Ces options sont précisées en
plaçant des ordres spéciaux dans un fichier d'initialisation nommé S-ICE.DAT.
S-ICE.DAT est un fichier texte ASCII que Soft-ICE parcourt au moment du
chargement. Ce fichier peut contenir des affectations de touches de fonction
une chaîne d'auto-démarrage et diverses options de configuration. Le fichier
peut être créé et édité avec tout traitement de texte DOS. Quand on charge
Soft-ICE depuis la ligne de commande, S-ICE.DAT doit être placé dans
le répertoire actuel ou dans un répertoire qui est accessible par votre
PATH actuel. Quand Soft-ICE est chargée comme un pilote de périphérique dans
CONFIG.SYS, S-ICE.DAT doit être dans le même répertoire que S-ICE.EXE.

Il y a trois catégories d'ordres qui peuvent être inclus dans le fichier
S-ICE.DAT:

* Les options spéciales de configuration
* Les affectations de touches de fonction
* Une séquence de commandes d'initialisation

6.4.1 Options Spéciales De configuration

Chacune des options suivantes de configuration qui pourraient être
nécessaire doivent chacune être placée sur une ligne différente dans
le fichier S-ICE.DAT.

* COMPAQ -- Les ordinateurs Compaq 386 et 386SX et quelque ordinateurs compatibles Compaq (y compris des ordinateurs contenant des cartes mère Micronix) ont 384K de mémoire étendue non-contigüe. L'option COMPAQ est nécessaire si vous voulez que Soft-ICE emploie cette mémoire.Notez que l'option COMPAQ est identique au paramètre de ligne de commande /C dans Soft-ICE 1.X.
* NOLEDS -- L'option NOLEDS indique à Soft-ICE ne pas allumer ni éteindre les LEDS du clavier pendant que la fenêtre Soft-ICE est activée. Certains claviers ont des problèmes de temporisation qui provoqueront des pertes de synchronisation avec le clavier. Si Soft-ICE s'arrête quand vous êtes dans sa fenêtre, employez cette option. Notez que l'option

NOLEDS est identique au paramètre de ligne de
commande /L dans Soft-ICE 1.X.

* NOTVGA -- L'option NOTVGA permet à Soft-ICE de fonctionner sur cartes compatibles VGA au niveau BIOS seulement. Beaucoup de cartes VGA ne sont pas compatibles avec le VGA IBM au niveau matériel. Ces cartes supportent le VGA au niveau BIOS seulement. Employez ce commutateur si vous avez un de ces adaptateurs vidéo. Notez que l'option NOTVGA est identique au paramètre de ligne de commande /V dans Soft-ICE 1.X.

* EXTENDED -- l'option EXTENDED force Soft-ICE à se charger directement en mémoire étendue sans message d'avertissement pour l'utilisateur. Il doit être employé si vous chargez Soft-ICE depuis le prompt du DOS et ne voulez pas être averti.Notez que l'option EXTENDED est identique au paramètre de ligne de commande /E dans Soft-ICE 1.X.

6.4.2 Affectations des touches de fonction

Une ou plusieurs commandes Soft-ICE peuvent être affectées à toute touche de fonction au moment du chargement. Voir la description de la commande FKEY dans la section 5.8 (Commandes de personnalisation du Débugger) pour une description de l'affectation des touches de fonction de la ligne de commande de Soft-ICE.

La syntaxe pour affecter une touche de fonction dans S-ICE.DAT est:

```
nom-de-la-touche-de-fonction = "chaîne"
nom-de-la-touche-de-fonction -- F1, F2... F12.
chaîne -- La chaîne peut consister en toute commande
               Soft-ICE valide et les caractères
               spéciaux ^ et ;.
               Un ^ est placé dans la chaîne pour rendre
               la commande invisible.
               Un ; placé dans la chaîne signifie un retour
               de chariot.
               La chaîne doit être entourée de guillements.
```

Un exemple d'affectation de touche de fonction dans S-ICE.DAT:

```
F12 = "D 100;"
```

Cela affectera la commande Soft-ICE DUMP à la touche F12. Quand F12 sera pressée Soft-ICE fera un dump de la mémoire à l'offset 100H dans le segment de données actuel. Le point virgule suivant le 100 représente la touche ENTREE.

6.4.3 Séquence de commandes d'initialisation

Une séquence de commandes peut être exécutée automatiquement au chargement de Soft-ICE. C'est utile pour personnaliser Soft-ICE comme vous le souhaitez. Par exemple, vous pouvez positionner des fenêtres et changer la séquence de touches d'activation.

La syntaxe pour positionner une séquence de commandes d'initialisation dans
S-ICE.DAT est:

```
INIT = "chaîne-d'affectation"
chaîne d'affectation -- La chaîne peut être constituée
                  de toute commande Soft-ICE valide et
                  des caractères spéciaux ^ et ;.
                  Un ^ est placé dans la chaîne pour
                  rendre une commande invisible.
                  Un ; placé dans la chaîne signifie un
                  retour de chariot.
                  La chaîne doit être incluse dans des
                  guillemets.
```

Un exemple séquence de commande d'initialisation dans S-ICE.DAT est:

```
INIT = "WIN; WR; WD 1; WC 12; ALTKEY CTRL X;"
```

Cet exemple mettra la fenêtre Soft-ICE en mode plein écran, crée une fenêtre
registre, crée une fenêtre données d'une ligne, crée une fenêtre code de 12
lignes, et change la séquence de touches de démarrage en CTRL X.

L'exemple S-ICE.DAT

Un exemple de fichier d'initialisation S-ICE.DAT est inclus sur la disquette
de distribution. Cet exemple affecte les touches de fonction

de la même façon que les touches de fonction dans le débugger Microsoft
CodeView.

Cet exemple S-ICE.DAT peut aussi être employé tel quel pour le didacticiel
du chapitre 3.

CHAPITRE 7

7.1 Introduction

Soft-ICE peut charger des programmes, des tables de symboles et des fichiers
source pour un débugging amélioré. Le débugging symbolique vous permet de
positionner des points d'arrêt et des variables de référence avec des
noms de symboles plutôt qu'en précisant des adresses numériques. Le débugging
niveau source vous permet d'avancer dans votre programme au niveau du code
source plutôt que dans le code assemblé.

Les numéros de ligne du source et les informations de symboles sont extraits
du fichier "MAP" de linkage.
Le fichier .MAP doit être compatible avec le linker Microsoft version
3.60 ou supérieur.

Les fichiers source et symboles sont stckés dans la mémoire étendue. Vous
devez avoir suffisament de mémoire étendue pour les fichiers source et
symboles. Les fichiers source ne sont pas chargés depuis le disque comme dans
beaucoup de débuggers. Cela permet à Soft-ICE de fournir un système complet de
débugging niveau source, vous pouvez débugger des T&SR, des routines
d'interruption et d'autres systèmes de code niveau source.

Note:
Vous ne pouvez pas utiliser le débugging symbolique ou niveau source si
Soft-ICE n'a pas été chargé comme pilote de périphérique dans
CONFIG.SYS.

7.2 Préparation pour débugging symbolique ou source

Avant de débugger un programme avec des symboles ou un source vous devez créer
un fichier de symbole. C'est un fichier binaire qui contient l'information de
numéros de ligne et de symboles dans un format que Soft-ICE peut comprendre.
Ce fichier est créé avec l'utilitaire MSYM.EXE. MSYM.EXE lit dans

votre
fichier .MAP pour créer un fichier de symbole avec l'extension (.SYM).

7.2.1 Préparation pour débugging symbolique seulement

Pour préparer un programme pour un débugging symbolique seulement, vous devez suivre les étapes suivantes:

1. Compiler ou assembler votre programme.
2. Linker votre programme avec les commutateurs adaptés pour créer le fichier .MAP qui contient une liste des symboles publics. Si vous utilisez le linker Microsoft, le commutateur à employer est /MA. Ce fichier doit être identique au fichier .MAP produit par le linker Microsoft, version 3.60 ou supérieure.
3. Créer un fichier .SYM en lançant MSYM.EXE. La syntaxe pour utiliser MSYM.EXE est:
   MSYM nom-programme [.extension]
   Si l'extension n'est pas indiquée MSYM suppose que l'extension est .MAP. MSYM lit dans un fichier MAP et écrit dans un fichier symbole. Le fichier symbole a le nom nom-programme.SYM.

Note:
Avant de compiler ou d'assembler votre programme vous pouvez vouloir rendre
un certain nombre de symboles publics. Seuls les symboles publics sont
supportés avec le débugging symbolique de Soft-ICE. La méthode pour rendre une
variable ou une étiquette publique varie, en fonction du langage que vous
utilisez.

En assembleur 8086, employez simplement la directive PUBLIQUE suivie par les
symboles locaux que vous désirez rendre publics. Par exemple:

```
PUBLIC FOO, LOOP1, STATUT
```

En langage C, tous les noms de procédure et les variables statiques définis
hors d'un bloc sont publics.

Pour autres langages, référez-vous au manuel de votre langage pour des détails.

7.2.2 Préparation pour débugging symbolique et source

Pour préparer un programme pour le débugging symbolique et source, vous devez
suivre les étapes suivantes:

1. Compiler ou assembler chaque module que vous désirez débugger au niveau source avec le commutateur approprié pour mettre les informations de numéros de ligne dans les fichiers objets. Avec les langages Microsoft langages vous pouvez employer les commutateur /Zi ou /Zd. Vous pouvez ne pas vouloir faire cela avec tous les fichiers, parce que les tailles combinées du fichier de symbole et de tous les fichiers source compilés avec ces commutateurs doivent rentrer dans la

mémoire étendue que vous avez réservée avec le commutateur /SYM dans CONFIG.SYS.

2. Linker votre programme avec les commutateurs propres à créer un fichier .MAP qui contient les numéros de ligne du source et une liste des symboles publics. Si vous utilisez le linker Microsoft, les commutateurs à employer sont /LI et /MA

   Le fichier .MAP doit être identique au fichier .MAP produit par le linker Microsoft, version 3.60 ou supérieure.

3. Créer un fichier .SYM en lançant MSYM.EXE. La syntaxe pour utiliser MSYM.EXE est:

   MSYM nom-programme [.extension]

Si l'extension n'est pas précisée MSYM suppose que l'extension est .MAP.
MSYM lit dans un fichier .MAP et écrit dans un fichier symbole. Le fichier symbole a le nom nom-programme.SYM.

7.3 Réservation de mémoire pour le fichier source et les symboles

Avant de charger des programmes, des fichiers source et des fichiers de symboles vous devez réserver de la mémoire étendue pour eux. La mémoire étendue est réservée quand vous chargez Soft-ICE dans CONFIG.SYS. Avant de réserver la mémoire étendue vous devez ajouter les tailles du fichier .SYM et de tous les fichiers source que vous voulez charger. Vous devez réserver au moins autant de mémoire étendue.
Vous devez employer le commutateur de chargement /SYM en chargeant S-ICE.EXE.
Une exemple de ligne dans CONFIG.SYS pour charger Soft-ICE et réserver de la place pour les fichiers source et symboles est:

```
DEVICE = S-ICE.EXE / SYM 1024
```

Cet exemple charge Soft-ICE en mémoire étendue et réserve 1 méga-octet de mémoire pour les fichiers source et symboles. Voir la section 6.3 (Chargement de Soft-ICE comme un pilote de périphérique) pour plus de détails sur les réservations de mémoire.

7.4 Chargement des Programmes et des fichiers symboles

L'utilitaire LDR.EXE est employé pour charger des programmes, des fichiers de symboles et des fichiers source. Pour faire du débugging symbolique sur des programmes d'application et des T&SR vous devez utiliser LDR.EXE pour charger le programme, les fichiers source et les symboles en une seule opération.
Pour débugger des pilotes de périphériques, des ROMS et autres composantes du

système vous emploirez LDR.EXE pour charger seulement le fichier symbole et
les fichiers source.

La syntaxe pour LDR.EXE est:

LDR nom-programme | nom-programme.SYM | nom-programme.extension

7.4.1 Chargement d'un programme, de symboles et de source

Pour charger votre programme, les fichiers source les symboles en une seule
opération, vous devez employer LDR.EXE sous cette forme:

    LDR nom-programme

Notez que nom-programme ne doit pas avoir d'extension. Si aucune extension
n'est fournie, alors LDR.EXE procèdera ainsi:

1. Il charge nom-programme.SYM en mémoire étendue
2. Il charge les fichiers source en mémoire étendue. Cette étape n'a lieu que s'il y a des enregistrements de source dans le fichier .SYM.
3. Il charge nom-programme.EXE en mémoire à l'emplacement où il aurait été chargé s'il avait été chargé directement du prompt DOS.
4. Il active Soft-ICE avec le pointeur d'instruction sur la première instruction de votre programme. S'il s'agit d'un programme en C et si le source est chargé pour le fichier contenant _MAIN, alors le source de ce fichier sera visible dans la fenêtre code.

7.4.2 Charger seulement les fichiers source et symboles

Si vous désirez charger seulement les fichiers source et symboles (pour
débugger un pilote de périphérique par exemple), vous devez employer LDR.EXE
sous la forme suivante:

    LDR nom-programme.SYM

Notez que the l'extension .SYM est précisée. Cela chargera le fichier .SYM et
les fichiers source en mémoire étendue. Quand les symboles sont chargés par
cette méthode vos symboles de pilote de périphérique ou de programme sont
supposés être référencés à 0:0. Comme c'est rarement le cas vous souhaiterez
utiliser la commande Soft-ICE SYMLOC pour reloger les symboles.
Voir la description de la commande SYMLOC dans la section 5.10 pour une
description complète.

Un exemple de chargement d'un fichier symbole appelé DRIVER.SYM:

    LDR DRIVER.SYM

7.4.3 Chargement d'un programme sans symboles ni source

Pour charger un fichier programme sans charger les symboles associés vous
devez employer LDR.EXE sous la forme:

```
  LDR nom-programme.extension
```

Notez que l'extension du fichier est présente. L'extension du fichier est .EXE
ou .COM. Quand une extension de fichier est précisée LDR.EXE charge le
programme et active Soft-ICE avec le pointeur d'instructions sur la première
instruction du programme.
Un exemple de chargement d'un programme avec des symboles et le source:

```
 LDR TEST.EXE
```

Notes:
LDR.EXE sauve automatiquement une copie de la table des vecteurs d'interruption
quand il charge votre programme. Celà équivaut à exécuter la commande VECS S.
Si vous êtes amenés à sortir de votre programme avant son achèvement, vous
devez faire un EXIT R pour sortir du programme et restaurer la table des
vecteurs d'interrumption.

L'utilisation de LDR.EXE pour charger seulement le nom-programme.EXE est
souvent utile pour redémarrer votre programme tandis que vous êtes au milieu
d'une séance de débugging. Pour redémarrer, la commande EXIT R pour terminer
la séance actuelle. Utilisez alors LDR.EXE pour recharger votre fichier .EXE.
Les symboles: le source ne doit pas être chargé puisqu'il reste en mémoire
étendue.

Si LDR.EXE vous affiche le message "Plus de place au chargement des
informations de symboles", cela signifie que vous ne réservez pas assez de
mémoire étendue avec le commutateur /SYM dans CONFIG.SYS. Si LDR.EXE ne
trouve pas vos fichiers source dans le même répertoire que le programme que
vous chargez, LDR.EXE vous demandera les noms de chemin où il peut trouver
les fichiers source. Si vous avez des fichiers source dans plusieurs
répertoires ou si vous chargez un programme fréquemment cela devient génant.
Vous pouvez éliminer ce prompt en utilisant la variable d'environnement DOS
SRC. LDR.EXE utilise cette variable d'environnement pour trouver les fichiers
source avant de demander à l'utilisateur.
La syntaxe pour positionner la variable d'environnement au prompt du DOS est:

```
   SET SRC = répertoire;répertoire;...;répertoire
```

Chacun des répertoires précisé sera exploré avant de demander à l'utilisateur.

Limitations:
Soft-ICE ne supporte les symboles que pour un programme à la fois. Si vous
chargez un nouveau fichier .SYM, le fichier existant est détruit.
Soft-ICE ne suit pas les overlays ou le mouvement des segments de Microsoft
Windows.
Soft-ICE reconnaît les symboles publics et les numéros de ligne seulement.
Il ne supporte pas les variables locales.

7.5 Débugging avec des symboles

Après avoir chargé votre programme et votre fichier .SYM vous pouvez commencer
le débugging symbolique de votre programme. En général un symbole peut être
utilisé dans toute commande à la place d'une adresse.

Les symboles sont aussi utilisés par certaines commandes Soft-ICE quand des
adresses sont affichées.
Par exemple, la commande U affiche les noms de symbole ou les labels et les
procédures comme il les rencontre.
Il y a deux commandes qui sont trés utiles quand vous faites du débugging
symbolique:

* SYM -- Utilisez la commande SYM pour obtenir une liste des noms de symboles et de leurs valeurs, ou pour changer la valeur d'un symbole.
* SYMLOC -- Utilisez la commande SYMLOC pour reloger la base de tous vos symboles. Vous utiliserez la commande SYMLOC quand:
    1. Vous chargez des symboles pour un pilote de périphérique
    2. Vous chargez des symboles pour un T&SR qui a déjà été chargé
    3. Votre programme se déplace lui-même à un emplacement autre que son emplacement original.

Voir la section 5. 10 pour une description complète de ces commandes.

7.6 Débugging avec source

Quand les fichiers source sont chargés, Soft-ICE vous permet de visualiser et
de vous déplacer dans le code source comme quand vous débuggez. Soft-ICE offre
deux modes différents de débugging niveau source: le mode mixte et le mode
source. Utilisez la commande SRC pour basculer entre ces deux modes.
Le mode mixte affiche les lignes de source et le langage d'assemblage produit
par ces lignes de source intercallés sur l'affichage. Le mode mixte est utile
quand vous devez débugger au niveau assembleur, mais utilise les lignes de
source comme référence. Le mode mixte est permis que la fenêtre de

code soit
visible ou non.

Le mode source ne montre que les lignes source sur l'affichage. Le débugging
niveau de source nécessite que la fenêtre de code soit visible.

7.6.1 Utilisation des numéros de ligne

Les numéros de ligne peuvent être utilisés à la place des adresses dans
plusieurs commandes. Pour différencier un numéro de ligne d'une adresse
réelle, placez un . (un point) avant le nombre. Par exemple, pour positionner
un point d'arrêt d'exécution à la ligne de source 45, entrez:

```
BPX .450
```

7.6.2 Utilisation du mode source dans la fenêtre code

La fenêtre code doit être visible pour basculer en mode source. Si elle n'est
pas visible, employez la commande WC pour la rendre visible. Une fois que vous
êtes en mode source vous pouvez employer les commandes Soft-ICE pour changer
vers un fichier source différent, visualiser un source à tout emplacement dans
le fichier, se déplacer à travers le fichier, rechercher des chaînes dans le
fichier, et positionner des points d'arrêt dans le fichier. Pour une
description complète des commandes suivantes voir leurs descriptions dans les
chapitres 4 et 5. La liste suivante est un bref aperçu des commandes qui sont
utiles quand vous débuggez du code source:

* Rendre la fenêtre code visible (si elle ne l'est pas déjà) avec la commande WC.
* Basculer entre les modes source, mixte, et code avec la commande SRC. Pour basculer entre les modes entrez:

  ```
  SRC
  ```

* Placer un fichier source dans la fenêtre code (s'il n'y est pas déjà) avec la commande FILE. Par exemple pour changer du fichier actuel pour le fichier MAIN.C entrez:

  ```
  FILE MAIN.C
  ```

* Afficher le source à un emplacement spécifique dans le fichier avec la commande U. Pour changer l'affichage à un numéro de ligne spécifique ou une adresse mémoire utilisez la commande U. Vous pouvez préciser l'adresse actuelle ou un numéro de ligne comme paramètre de la commande. Par exemple, pour voir le source dans la fenêtre de code en commençant à la ligne de source 450 entrez:

  ```
  U .450
  ```

* Trouver l'instruction actuelle dans la fenêtre code avec la commande . (le point).

* Rechercher une chaîne de caractères particulière avec
  la commande S. Par exemple, pour chercher la chaîne
  "Bonjour Monde" en commençant à ligne 100 dans le
  fichier source actuel entrez:

```
SS 100 "Bonjour Monde"
```

* Déplacer le curseur vers la fenêtre de code
  (s'il n'y est pas déjà) avec la commande EC.
* Faire défiler le source avec les touches haut,
  bas, PaqeUp, PageDn.
* Postionner des points d'arrêt avec la commande
  BPX. Placer simplement le curseur sur la ligne
  de source où vous désirez positionner le point
  d'arrêt, entrez alors:

```
BPX
```

CHAPITRE 8

8.1 Introduction

Soft-ICE a un gestionnaire de mémoire paginée dans son noyau. Le gestionnaire de mémoire paginée de soft-ICE supporte les spécifications Lotus-Intel-Microsoft 4.0. Cette caractéristique de Soft-ICE est utile si vous utilisez des programmes qui supportent les spécifications EMM, ou si vous devez "backfiller" votre mémoire conventionnelle pour l'étendre au-delà de 640K.

D'autres programmes de contrôle 386 qui fournissent des possibilités EMM (tels que QEMM ou 386MAX) ne coexisteront pas avec Soft-ICE. Si vous utilisez ces programmes pour un support EMM ou du "backfilling", vous pouvez employer le gestionnaire EMM de Soft-ICE à leur place.

Activer les capacités EMM de Soft-ICE entraîne les mesures suivantes:

1. Configure l'environnement de mémoire avec l'utilitaire EMMSETUP.EXE. Cette utilitaire modifie S-ICE.EXE avec la carte des pages EMM désirée.
2. Ajouter le commutateur /EMM à votre ligne S-ICE.EXE dans votre CONFIG.SYS. Cela réserve une fraction de la mémoire étendue pour la mémoire paginée. Une ligne d'exemple de CONFIG.SYS qui réserve de la mémoire pour EMM:
   ```
   DEVICE = S-ICE.EXE /EMM 2048
   ```
   Cela réservera 2 méga-octets de mémoire étendue pour l'usage de EMM. Voir la section 6.3 (Chargement Soft-ICE comme un Pilote de périphérique) pour des détails de l'installation de Soft-ICE dans CONFIG.SYS.
3. Rebootez votre système.

8.2 Configuration de l'Environnement EMM

Avant d'installer S-ICE.EXE avec le commutateur /EMM dans CONFIG.SYS, vous devez lancer EMMSETUP.EXE pour configurer l'environnement EMM

4.00.
Ce procédé de configuration vous permet de sélectionner quelles portions
de mémoire vous aimeriez rendre disponible comme pages EMM 4.0. Lancer
EMMSETUP.EXE est hautement recommandé si vous utilisez des programmes qui
prennent avantage des spécifications EMM 4.0.

8.2.1 Pages EMM par Défaut

Par défaut, S-ICE.EXE avec le commutateur /EMM est pre-configuré pour permettre des pages EMM 4.0 dans les zones de mémoire suivantes:

* Les 640K inférieurs (à l'exception des 1er 64K)
* Les 64K commençant à DDH

Vous pouvez vouloir reconfigurer pour l'une des raisons suivantes:

* Vous pouvez avoir un pilote tel qu'un réseau qui utilise la zone mémoire de D000H.
* Vous pouvez vouloir remplir plus de trous au-dessus de 640K avec des pages EMM. Cela accroîtra la performance et la convivialité de programmes comme Microsoft Windows. Pour obtenir la performance maximum de Microsoft Windows, vous devriez remplir chaque page disponible avec de la mémoire paginée accrue.

8.2.2 Personnalisation de la carte des pages EMM

Pour configurer la carte EMM vous devez employer l'utilitaire EMMSETUP.EXE.
EMMSETUP.EXE permet de modifier la carte des page, puis met à jour S-ICE.EXE
avec les changements.

EMMSETUP fait sa meilleure supposition en configuration automatiquement de la
carte EMM. EMMSETUP essayera de remplir le plus d'espace possible avec des
pages disponibles autour des cartes de vidéo et des ROMS. Si sa supposition
n'est pas bien assez ou de ne vous satisfait pas tout à fait, vous pouvez
l'outrepasser. L'outrepasser peut être nécessaire si vous avez un réseau ou
un adaptateur vidéo spécial.
Pour configurer la carte EMM entrez:

```
EMMSETUP
```

EMMSETUP affiche une matrice de pages de 16K disponibles dans le méga-octet
inférieur. La matrice est partagé en 16 colonnes, chacun représentant 64K
(de 0 à 10000H). Il y a 4 rangées représentant les quatre pages de 16K dans
chaque région de 64K.

Chaque bloc de la matrice peut contenir un E, un X, un R ou un V. Les blocs
contenant un E sont disponibles comme pages EMM; les blocs contenant un X
ne le sont pas. Les blocs contenant un R sont des zones de mémoire qui ont
été identifiées par EMMSETUP comme zones de ROM. Vous pouvez marquer

ces
zones avec un E si vous le désirez, cependant, cela devrait seulement être
fait si la ROM n'est jamais accédée. Les blocs contenant un V sont identifiés
comme mémoire vidéo. Nous avons fait des suppositions très larges sur la mémoire
vidéo. Votre carte vidéo ne devrait pas prendre autant que nous avons réservé.
Vous pouvez outrepasser la mémoire qui semble contenir des V inutiles si vous
le désirez.

Si vous êtes satisfait des suppositions D'EMMSETUP, pressez F10 et S-ICE.EXE
sera modifié avec ces paramètres. Vous devez rebooter avant que tout
changement fait à S-ICE.EXE prenne effet. Si vous désirez modifier les
suppositions D'EMMSETUP, le faire maintenant.

8.2.2.1 Inclure et Exclure des zones de EMM

Pour inclure une zone comme mémoire EMM 4.0 guidez simplement le curseur sur
le bloc désiré, tapez alors E. Inversement, pour exclure une zone de la mémoire
EMM 4.0, guidez le curseur sur le bloc et tapez X. Quand vous êtes satisfaits
de vos changements, pressez F10 pour sortir du programme. Tous les changements
sont automatiquement sauvés dans le fichier S-ICE.EXE. Si vous désirez sortir
sans modifier S-ICE.EXE pressez ESC.

Vous devez rebooter avant que les changements faits à S-ICE.EXE ne prennent
effet.

Quand vous incluez de la mémoire supérieure gardez à l'esprit les faits
suivants:

* CGA occupe de B800H à C000H.
* MDA occupe de B000H à B100H.
* Les cartes Hercule occupent de B000 à C000H.
* EGA occupe de A000H à C000H et de C000H à C400H.
* VGA (sur carte mère) occupe de A000H à C000H.
* VGA (sur carte d'extention) occupe de A000H à C000H et C000H à C800H.
* PS / 2 système en ROM occupe de E000H à 10000H.
* PS / 2 ESDI en ROM occupe de CC00H à D000H
* La plupart des Roms de compatibles AT occupent de F000H à 10000H.
* Les systèmes Compaq, les cartes mères Micronix et beaucoup de carte mères Chips et Technologies déplacent la ROM EGA/VGA à E000H. Cependant ils occupent encore la région C000H.
* Les réseaux 'Token Ring' occupent habituellement de CC00H à E000H.
* Beaucoup de réseaux occupent des régions de mémoire dans la zone D000H.

Les principes ci-dessus sont valables pour des cartes 'normales', beaucoup
d'utilisations par les fabricants d'ordinateurs et par les fabricants de
cartes diffèrent.

8.3 Autres caractéristiques de l'EMM

S-ICE.EXE avec le commutateur /EMM a deux caractéristiques qui sont automatiquement activées suivant votre configuration système. Ces caractéristiques sont: le 'backfilling' et le relogement du cadre de page.

8.3.1 Accroitre la Mémoire Conventionnelle

La mémoire système peut automatiquement être 'backfilled' jusqu' a la première page non-mappable. Cela signifie que l'on commence à chercher des E contigus à partir de l'emplacement 1000, et qu'on continue jusqu'à trouver le premier E non-contigu. Si les E contigus vont au-delà de la mémoire de base de votre système, la mémoire est 'backfilled' jusqu'au premier R, V, ou X qui est trouvé.

Le bénéfice de 'backfilling' est que vous pouvez accroitre la quantité de mémoire utilisable de système à plus de 640K. La mémoire 'backfilled' est disponible sous DOS. Si vous ne voulez pas de mémoire 'backfilled', employez EMMSETUP pour rendre les pages non-mappable (X) à l'endroit ou vous désirez que se termine votre mémoire système.

Note:
Les systèmes uniquement monochromes (MDA) peuvent 'backfill' jusqu'à B000H pour ajouter un supplément de 64K à la mémoire conventionnelle; les systèmes CGA peut être 'backfilled' jusqu'à B800, pour ajouter un supplément de 96K à la mémoire conventionnelle. Les systèmes EGA et VGA peuvent être 'backfilled' seulement si aucun programme graphique ne doit êtrre exécuté.

Avertissement:
Si la mémoire est 'backfilled', NE DÉCHARGEZ PAS Soft-ICE. Le faire planterait votre système.

8.3.2 Localisation Automatique du cadre de page

Beaucoup de programmes connus requièrent un cadre de page de 64K inemployé comme mémoire DOS normale. C'est normalement situé au-dessus de la zone de la mémoire vidéo. Cependant dans certains systèmes il y a aucune zone de 64K contigus pour placer le cadre de page. Dans ce cas, S-ICE.EXE 'vole' 4 pages mappables en mémoire basse. Le résultat net est que la mémoire DOS rétrécit de 64K.

8.4 Débugging sous EMM

Un point d'arrêt domaine ou un point d'arrêt mémoire qui est dans une zone
EMM restera à cette adresse que la page soit mappée ou non.

Quand vous débuggez des programmes EMM, la commande EMMMAP peut aussi être
très utile. Voir la section 5.6 pour plus d'information.

Les commandes D, E, S, F, et C peuvent être employés voir ou modifier nimporte
quelle page EMM affectée. La page ne doit pas à être actuellement mappée.
La syntaxe de ces commandes est similaire à celle des commandes qui étaient
employées pour les pages non-EMM, à l' exception de la suivante:

* Dans les commandes D, E, S, et F, la partie adresse
  de la commande doit être précisée de la façon suivante:
  Hhandle# Ppage# décalage
  où handle est un nombre précisant quel handle EMM employer,
  page est un nombre précisant quelle page EMM à employer,
  décalage est un nombre de 0 à 4000H, précisant le décalage
  par rapport au début la page.

Exemple:

```
DB H1 P3 0
```

Cette commande déchargera les octets de la page 3 du
handle 1, en commençant au décalage 0.

* La commande C doit être précisée de la façon suivante:
  C Hhandle# Ppage# offset1 Llength offset2
  Où handle et page sont comme ci-dessus.
  offset1 est un nombre de 0 à 4000H, précisant
  le décalage par rapport au début de la page,
  où le premier bloc de données à comparer est situé.
  offset2 est un nombre de 0 à 4000H, précisant
  le décalage par rapport au début de la page,
  où le deuxième bloc de données à comparer est situé.

Exemple:

```
C H2 P4 00 L10 1000
```

Cette commande comparera les 10 premiers octets de mémoire situés
au décalage 0 de la page 4 du handle 2 avec les 10 premiers octets
de mémoire situés au décalage 1000 de la page 4 du handle 2.

Note: Les emplois suivants des commandes D, E, S, F, et C continueront à
employer le handle et la page précisés en dernier. Pour revenir à la mémoire
conventionnelle, utilisez un des ordres ci-dessus avec un segment précisé
dans le champ adresse, par exemple:

```
D 0:0
```

CHAPITRE 9

Les domaines de trace arrière

9.1 Introduction

Soft-ICE peut enregistrer les informations d'instructions dans un buffer de trace arrière pendant que votre programme s'exécute. Ces instructions peuvent alors être affichées après qu'un bug se soit produit. Cela vous permet de remonter en arrière et de retracer l'action d'un programme pour déterminer le flux réel d'instructions précédant un point d'arrêt.

Les informations d'instructions sont recueillies sur des accès dans un domaine d'adresse spécifié, plutôt que dans tout le système. Les domaines peuvent être de 1 octet à 1 méga-octet, ainsi, si vous le souhaitez, vous pouvez obtenir une information complète sur le système. L'utilisation de domaines spécifiques plutôt que recueillir toutes les instructions est utile pour deux raisons:

1. Le buffer de trace arrière n'est pas encombrée par des informations qui ne vous intéressent pas. Par exemple, vous pouvez ne pas être intéressés par l'activité des interruptions ou l'exécution de MSDOS.
2. Les domaines de trace arrière dégradent les performances du système lorsqu'ils sont actifs. En limitant le domaine à une zone qui vous intéresse, vous pouvez améliorer énormément les performances de votre système.

Soft-ICE a deux méthodes pour utiliser les instructions dans le buffer de trace arrière:

1. La commande SHOW vous permet d'afficher les instructions du buffer de trace arrière. Vous devez préciser de combien d'instructions vous désirez remonter dans le buffer.
2. La commande TRACE vous permet de remonter et de 'rejouer' les instructions du buffer de trace arrière, cette méthode vous permet de voir le flux d'instruction dans le contexte du code source ou du code du programme.

9.2 Utilisation des domaines de trace arrière

Pour employer les domaines de trace arrière vous devez procéder

ainsi:

1. Affecter un buffer de trace arrière de la taille désirée en insérant le commutateur /TRA sur la ligne S-ICE.EXE dans CONFIG.SYS. Par exemple, pour créer un buffer de trace arrière de 100K vous devez avoir la ligne suivante dans votre fichier CONFIG.SYS:
   DEVICE = S-ICE.EXE 100
   Un buffer de trace arrière de 10K est affectée par défaut. Si c'est suffisant pour vos besoins vous ne devez pas affecter un plus grand buffer.
   La taille du buffer d'historique est seulement limitée par la quantité de mémoire étendue disponible.
2. Permettre des domaines de trace arrière en créant un point d'arrêt doamine mémoire avec le verbe T ou TW. Par exemple:
   BPR 1000:0 2000:0 T
   Les verbes T et TW ne provoquent pas de point d'arrêt, mais ils notent les informations d'instructions qui peuvent être affichées ultérieurement avec les commandes SHOW ou TRACE.
3. Positionnez tout autre point d'arrêt si vous le désirez.
4. Sortez de Soft-ICE avec la commande X.
5. Après qu'un point d'arrêt se soit produit, ou après que vous ayez activé Soft-ICE, vous pouvez afficher les instructions dans le buffer avec la commande SHOW. Par exemple, pour remonter de 50 instructions dans le buffer et afficher les instructions entrez:
   SHOW 50

6. Pour 'rejouer' une série d'instructions vous devez d'abord entrer en mode simulation de trace avec la commande TRACE. Pour commencer à rejouer la séquence d'instructions commençant 50 instructions en arrière dans le buffer entrez:

   TRACE 50
7. Après être entré en mode simulation de trace, vous pouvez tracer la séquence d'instructions en utilisant les commandes XT, XP, ou XG. Cela vous permet de retrouver le flux de programme. Par exemple, vous pouvez exécuter pas à pas la séquence d'instructions dans le buffer, en commençant à l'instruction précisée par la commande TRACE en entrant:

   ```
   XT
   XT
   .
   .
   .
   XT
   ```

   La commande XT trace pas à pas dans le buffer de trace arrière. La commande XP trace dans le buffer de trace arrière. La commande XG va à une adresse dans le buffer de trace arrière.
8. Pour sortir du mode simulation de trace entrez:

   TRACE OFF
9. Pour remettre à 0 le buffer de trace arrière, utilisez la commande X.

9.3 Notes spéciales

Lorsque vous êtres en mode simulation de trace, beaucoup de commandes Soft-ICE fonctionnent comme d'habitude, y compris lorsqu'on affichez la carte mémoire ou lorsque l'on édite des données. Les exceptions sont:

1. L'information de registre n'est pas sauvée dans le buffer de trace arrière, donc les valeurs des registres ne changent pas lorsque vous tracez dans le buffer, à l'exception de CS et de IP.
2. Les commandes qui provoquent normalement la sortie de Soft-ICE ne fonctionnent pas dans le mode simulation de trace. Ce sont X, T, P, G, EXIT.

Pendant que vous lisez des instructions dans le buffer de trace arrière avec les commandes TRACE et SHOW, vous pouvez constater des singularités dans l'exécution des instructions. Elles sont causées par des sauts dans et hors du domaine spécifié. Ils surviennent habituellement lors des sauts, appels, retours et points d'entrée. Quand vous avez un problème de plantage ou ou autre difficulté qui requiert des domaines de trace arrière, vous devrez souvent employer des domaines très étendus pour arriver à circonscrire le problème. Une fois que vous avez une meilleure idée de la zone spécifique du problème, vous pouvez utiliser des domaines plus petits.

Les grands domaines de trace arrière sont souvent très lents. Quand on utilise des domaines étendus, c'est habituellement pour essayer d'obtenir une idée générale de l'endroit ou se trouve le problème. Soft-ICE a un mode spécial pour traiter de grands domaines. Cela accélère les domaines d'un facteur trois ou plus, mais limite la quantité d'instructions dans le buffer.

Ce mode recueille seulement les instructions qui écrivent dans la mémoire dans un domaine spécifié.

Ce mode fonctionne mieux avec des grands domaines et tend à être moins efficace pour de petits domaines. Pour permettre de genre de domaine de trace arrière, employez la commande BPR avec le verbe TW au lieu de T. Par exemple:

```
 BPR 1000:0 2000:0  TW
```

Pour davantage d'informations sur les domaines de trace arrière voir les descriptions des commandes:

SHOW, TRACE, XT, XP, XG, XRSET, BPR

CHAPITRE 10

10.1 Introduction

MagicCV vous permet d'utiliser Microsoft CodeView dans moins de 8K de mémoire
conventionnelle sur votre 80386.

MagicCVW vous permet d'utiliser Microsoft CodeView pour Windows dans moins de
8K de mémoire conventionnelle sur votre 80386.

L'utilisation de Soft-ICE en combinaison avec MagicCV ou MagicCVW vous permet
d'avoir la puissance de Soft-ICE pendant que vous avez la facilité d'utiliser
les produits CodeView avec lesquels vous êtes familiers.

Dans le reste de ce chapitre, le terme MCV s'appliquera à MagicCV ET MagicCVW,
et le terme CV s'appliquera à CodeView et CodeView pour Windows.

10.2 Fonctionnement de Soft-ICE avec MagicCV ou MagicCVW

Pour utiliser Soft-ICE 2.0 et MCV ensemble, vous devez installer S-ICE.EXE
comme un pilote de périphérique. S-ICE.EXE est fourni sur la disquette
Soft-ICE. S-ICE.EXE remplace NUMEGA.SYS dans CONFIG.SYS. Utiliser les
commutateurs /MCV, /EMM, et /EXT commutateurs comme si vous utilisiez MagicCV
ou MagicCVW seul. Il y a des commutateurs supplémentaires que vous pouvez
vouloir utiliser pour Soft-ICE. Se référer au chapitre 6 pour des informations
sur ces commutateurs.

Pour lancer MagicCV ou MagicCVW après que Soft-ICE ait été chargé, référez-vous
à votre manuel de MagicCV ou de MagicCVW.

Note:
MagicCVW nécessite Soft-ICE version 2.00 ou supérieure.

MagicCV nécessite Soft-ICE version 1.02 ou supérieure. Les pilotes S-ICE.SYS
et NUMEGA.SYS étaient fournis avec certaines versions de Soft-ICE. Les
pilotes S-ICE et NUMEGA

doivent être remplacés par S-ICE.EXE avant de pouvoir lancer MagicCV et
Soft-ICE 2.0 ensemble.

10.3 Considérations Spéciales

Deux Machines Virtuelles

Quand vous utilisez Soft-ICE et MCV ensemble, vous devez garder a l'esprit
que CV est dans une machine virtuelle séparée de l'environnement cible.
Vous pouvez activer Soft-ICE depuis l'une ou l'autre machine virtuelle, par
exemple, quand CV fonctionne, ou quand le programme cible fonctionne.

Si vous activez Soft-ICE tandis que le programme cible fonctionne, tout
se passe comme définit dans le manuel de Soft-ICE. Si vous activez Soft-ICE
tandis que CV fonctionne (fait typiquement pour les points d'arrêt), vous
devez garder quelques points à l'esprit:

* Les registres sont ceux de CV et ils NE PEUVENT PAS être changés.
* Par facilité, la commande Soft-ICE MAP affiche la carte mémoire de la machine virtuelle du programme cible, pas la carte mémoire de la machine virtuelle de CV. La zone en surbrillance dans la carte mémoire peut ne pas être correcte.
* Tout affichage ou modification de mémoire se produit dans la machine virtuelle du programme cible.
* Vous n'avez aucune visibilité dans la machine virtuelle de CV à l'exception de l'affichage des valeurs de registre. Rappelez-vous que quand vous activez la fenêtre Soft-ICE tandis que CV est actif, les valeurs de registre sont celles de CV et ne doivent pas être modifiées.
* La trace des programmes et des instructions est inactivée de la fenêtre Soft-ICE quand CV est actif. Ceci doit empêcher toute confusion, parce qu'une trace s'effectuerait alors dans CV, pas dans le programme cible.
  Si vous essayez faire une Trace Soft-ICE (T) ou un Pas De Programme (P) tandis que CV est actif, vous obtiendrez le message d'avertissement: "Function not available in CV virtual machine." Pour tracer dans le code de votre programme cible, vous pouvez utiliser une de ces deux options:
  * Utiliser la commande de trace de CV. Pour faire cela, quitter la fenêtre Soft-ICE en utilisant la commande X, faites alors une ou plusieurs traces CV pour balayer le programme cible.
  * Utiliser Soft-ICE pour aller à l'adresse du programme cible, utiliser alors la commande Soft-ICE T ou P pour balayer votre programme cible. Pour cela, quitter la fenêtre Soft-ICE avec la commande X, pressez alors la touche 'F3' jusqu'à ce que CV soit en mode mixte. Cela vous permet de voir les lignes de source et les adresses d'instruction. Activer Soft-ICE. Si la fenêtre Soft-ICE n'est pas déjà en mode NARROW, utilisez la commande WIN pour changer la taille de la fenêtre. Déplacer la fenêtre Soft-ICE fenêtre pour pouvoir voir

l'adresse d'instruction sur le côté gauche de l'écran. Maintenant vous pouvez utilier la commande Soft-ICE G pour aller à l'une des adresses. Soyez sûr de taper l'adresse entière, (segment et décalage). Entrer alors 'G' dans la fenêtre CV. A ce point, CV n'est pas actif, et vous pouvez employer les commandes Soft-ICE T ou P pour balayer le programme cible.

La commande SHELL de CodeView

Si vous utilisez le DOS Shell dans la machine virtuelle de CodeView, le DOS Shell fait partie de la machine virtuelle. A cause de cela, vous ne devez pas lancer de T&SR depuis le DOS Shell. Si vous le faites, quand vous sortez de CodeView le T&SR disparaîtra avec la machine virtuelle.

C'est dangereux, parce que nimporte quel vecteur d'interruption non restauré pourrait planter votre machine.

Le commutateur /R de CV

Soft-ICE tire parti de beaucoup des caractéristiques du 80386 incluant les registres de débugging du 80386. Cela signifie que les registres de débugging ne sont pas disponibles pour CV, et vous ne pouvez pas employer le commutateur /R de CV quand quand Soft-ICE fonctionne. Si vous employez le commutateur /R, Soft-ICE vous donne une erreur générale de protection. A ce point, vous pouvez presser "C" pour continuer, relancer CV sans le commutateur /R, et utiliser les points d'arrêt de Soft-ICE.

Le commutateur /R de CV fonctionne quand vous lancez MCV sans Soft-ICE.

3.4 La commande Soft-ICE ACTION

La commande ACTION offre trois méthodes différentes pour activer CV après un point d'arrêt Soft-ICE. Le meilleur choix d'action est ACTION NMI. Si vous avez des problèmes avec ACTION sur NMI (habituellement parce qu'une carte dans votre système utilise NMI),utilisez ACTION INT1.

SECTION III - Caractéristiques

CHAPITRE 11

Caractéristiques Avancées

11.1 Utilisation de Soft-ICE avec d'autres débuggers

Soft-ICE est conçu pour bien fonctionner avec d'autres débuggers. Chaque débugger offre des caractéristiques différentes, et peut donc nécessiter un traitement spécial. Cette section décrira certaines méthodes pour employer plusieurs débuggers efficacement.

11.1.1 Débuggers utilisant DOS

Beaucoup de débuggers utilisent DOS et la ROM BIOS pour leurs affichages et leurs entrées clavier. Une attention particulière doit être apportée quand on utilise ces débuggers avec Soft-ICE (ex: DEBUG, SYMDEB, et CODEVIEW), parce que DOS et la ROM BIOS ne sont pas pleinement ré-entrants. Si un point d'arrêt survient tandis que du code s'exécute dans le DOS ou le BIOS, un problème de ré-entrance peut survenir.

Soft-ICE fournit un avertissement optionnel de ré-entrance, qui est activée avec la commande WARN. Quand le mode WARN est ON, Soft-ICE surveille une éventuelle ré-entrance du DOS ou de la ROM BIOS avant de générer l'ACTION qui active le débugger hôte. Quand un problème de ré-entrance est détecté, Soft-ICE affiche un message d'avertissement et vous offre le choix entre continuer à exécuter le code ou revenir à Soft-ICE.

Notez que Soft-ICE lui-même n'emploie pas DOS ou la ROM BIOS dans ses

commandes de débugging. Cela signifie que vous pouvez employer Soft-ICE tout le temps, sans problèmes de ré-entrance.

Pour plus d'information sur la commande WARN, voir la section 5.4.

11.1.2 La commande ACTION avec d'autres débuggers

Différents débuggers utilisent des méthodes différentes pour s'activer. Pour une description de ces méthodes voir la section 13.1.

Si vous voulez revenir à votre débugger après avoir atteint un point d'arrêt, vous devez changer ACTION (voit la section 5.4) pour travailler avec votre débugger.

Dans la plupart des cas, l'action qui doit être utilisée après qu'un point d'arrêt soit atteint est INT3. Par exemple, DEBUG et SYMDEB fonctionnent avec ACTION réglé sur INT3.

Si INT3 ne fonctionne pas avec votre débugger, essayez INT1 ou NMI. CODEVIEW fonctionne avec ACTION réglé sur NMI.

11.1.3 Considérations Spéciales

Quand un point d'arrêt est positionné, vous devez prendre garde à ne pas le désactiver par mégarde. Par exemple, si vous établissez un point d'arrêt mémoire à 0:0, puis utilisez votre débugger pour visualiser l'emplacement mémoire 0:0, Soft-ICE sera activé. Si ACTION est positionné pour activer votre débugger, alors votre débugger sera déclenché par lui-même. Comme certains débuggers ne sont pas ré-entrants, cela peut être un problème fatal.
Ce problème peut aussi survenir avec d'autres fonctions de débugging, comme l'édition ou le désassemblage.

Pour cette raison, il est préférable de désactiver les points d'arrêt Soft-ICE lorsque Soft-ICE vous a aidé à atteindre le point que vous vouliez examiner avec votre débugger.

11.1.4 Utilisation de Soft-ICE avec CODEVIEW

Soft-ICE travaille mieux avec CODEVIEW quand CODEVIEW est en mode assembleur ou en mode mixte. Quand CODEVIEW est en mode source avec des langages de haut niveau les points d'arrêt ne fonctionnent pas toujours correctement.
Il est toujours préférable d'utiliser ACTION NMI quand vous voulez que Soft-ICE active CODEVIEW.

11.1.5 Débuggers utilisant les points d'arrêt registre du 80386

Le 80386 a 4 points d'arrêts registre disponibles pour l'utilisation par des
débuggers. Soft-ICE les utilise pour ses points d'arrêt mémoire octets, mot et
double mot. Si le débugger que vous utilisez avec Soft-ICE les utilise aussi,
il y aura un conflit. Il y a deux manières de résoudre ce problème.

1. Désactiver l'utilisation des points d'arrêt registre du 80386 dans le débugger que vous utilisez avec Soft-ICE. Vérifier la documentation de votre autre débugger pour une description de cette fonction.
2. Certains débuggers emploient automatiquement les registres de point d'arrêt s'ils détectent un processeur 80386 sans aucune méthode pour les désactiver (certaines versions de SYMDEB procèdent ainsi). Pour ces débuggers procéder ainsi:
   * Activer la fenêtre Soft-ICE avant de lancer l'autre débugger.
   * Activer le mode BREAK de Soft-ICE avec la commande BREAK (vous pouvez faire cela dans le la commande INIT de S-ICE.DAT si vous faites cela fréquemment).
   * Lancez votre autre débugger.
   * Vous pouvez maintenant activer la fenêtre Soft-ICE et mettre le mode BREAK OFF si vous le souhaitez.

11.2 Points d'arrêt utilisateur

Occasionnellement vous pouvez avoir besoin d'un ensemble très particulier de
conditions de point d'arrêt. Si des conditions spéciales nécessitent de
calculer des valeurs de registre ou des valeurs de mémoire, vous pouvez
écrire une routine de calcul de point d'arrêt.

Soft-ICE contient un mécanisme très général pour appeler des routines de
calcul de points d'arrêt utilisateur: la commande ACTION. Quand vous utilisez
la commande ACTION, Soft-ICE peut acheminer tous les points d'arrêt vers
un vecteur d'interruption spécial. Cependant, avant que les points d'arrêt
puissent être acheminés, la routine de calcul doit être placée en mémoire, et
le vecteur d'interruption doit pointer vers la routine de calcul.

Tous les registres ont les mêmes valeurs que quand un point d'arrêt Soft-ICE
survient. Il est de la responsabilité de la routine de calcul de sauvegarder et
de restaurer les registres. Si votre routine de calcul détecte une concordance
avec des conditions de point d'arrêt, elle peut faire une quantité d'actions.
Certains exemples d'actions utiles qu'une routine peut faire quand une
concordance est trouvée sont:

* stocker l'information pour plus tard
* envoyer l'information directement vers une imprimante ou un terminal sériel

* émettre une instruction INT3 pour activer Soft-ICE.
  La commande 13HERE doit être positionné sur ON pour
  que l'INT3 active Soft-ICE (voit la section 5.4).

Si les conditions ne concordent pas, la routine de calcul doit exécuter une
instruction IRET. Pour résumer:

1. Créer une routine de calcul de point d'arrêt dans votre espace code, ou n'importe où dans la mémoire libre. La routine doit préserver les registres. Après avoir comparé les conditions désirées, la routine peut exécuter soit une INT3 pour activer Soft-ICE, soit un IRET pour continuer.
2. Faire pointer un vecteur d'interruption inutilisé vers votre routine de calcul. Cela peut être fait dans votre code ou depuis Soft-ICE.
3. Dans Soft-ICE, positionner ACTION au numéro d'interruption qui a été employé pour pointer sur votre routine de calcul.
4. Dans Soft-ICE, positionner 13HERE sur ON. Ceci est nécessaire pour activer Soft-ICE après que les conditions aient été rencontrées.
5. Positionner les conditions générales de point d'arrêt de Soft-ICE. Quand l'une de ces conditions de point d'arrêt est rencontrée, votre routine de calcul sera appelée.

11.2.1 Exemple d'un point d'arrêt utilisateur

Cette section contient un exemple d'un point d'arrêt utilisateur qui calcule
pour les conditions AX = 3, BX = 4 et CX = 5 quand un point d'arrêt se produit.

En premier, nous créons la routine de calcul. Pour les besoins de cet exemple,
nous assemblerons la commande directement en mémoire avec l'assembleur
interractif de Soft-ICE. Pour cet exemple nous assemblerons arbitrairement la
routine à l'emplacement 9000:0H. Les séquences suivantes sont entrées dans
Soft-ICE:

```
A 9000:0
9000:0 CMP AX, 3
9000:3 JNE 10
9000:5 CMP BX, 4
9000:7 JNE 10
9000:A CMP CX, 5
9000:D JNE 10
9000:F INT3
9000:10 IRET
```

Maintenant que la routine est en mémoire, vous devez faire pointer un vecteur
d'interruption vers la routine. Pour cet exemple, nous prendrons
arbitrairement INT 99H. Pour placer 9000:0H dans le vecteur INT 99H
entrez:

```
ED 0:99*4 9000:0
```

Positionner la coomande ACTION de telle façon que Soft-ICE appelle votre
routine de calcul de point d'arrêt à chaque point d'arrêt.

```
ACTION 99
```

Positionner 13HERE sur ON; la routine de calcul peut activer Soft-ICE quand
les conditions surviennent.

```
13HERE ON
```

Maintenant vous devez positionner les points d'arrêt. Pour cet exemple, nous
sommes justes intéressés quand les registres sont: AX = 3, BX = 4, CX = 5
dans un programme spécifique, et nous ne voulons pas d'autre calcul. Pour
faire cela, on emploie un point d'arrêt registre mémoire lecture:

```
BPR segment:début-offset segment:fin-offset
```

Cela provoquera l'appel de votre routine de calcul de point d'arrêt après
l'exécution de chaque instruction dans le domaine mémoire précisé. Quand les
conditions de registre ne correspondent pas, alors l'instruction IRET est
exécutée. Quand les conditions correspondent aux calculs précisés, l'INT 3
est exécutée et Soft-ICE est activé.

Quand Soft-ICE est activé, le pointeur d'instruction indiquera l'INT3 dans
votre routine de calcul (9OOO:FH dans notre exemple). Pour arriver à
l'instruction suivant celle qui a causé le point d'arrêt, vous devez changer
le pointeur d'instruction pour indiquer l'instruction IRET (F000:10H dans
l'exemple) et tracer un pas. C'est accompli avec les commandes suivantes

```
RIP IP + 1
T
```

Après que vos conditions de point d'arrêt aient disparu, pensez à changer la
commande ACTION pour ACTION HERE afin que les points d'arrêts suivants
n'aillent pas dans votre routine de calcul.

11.3 La Fenêtre en mode Graphique

L'écran est basculé en mode texte quand Soft-ICE est activé. Si l'écran était
en mode graphique ou en mode 40 colonnes, l'affichage graphique n'est
pas visible pendant que la fenêtre est active. Pour les utilisateurs qui
doivent voir les affichages graphiques en débugging, trois caractéristiques
sont fournies. La première caractéristique permet à la fenêtre Soft-ICE
d'afficher sur un deuxième moniteur (voir la commande ALTSCR, section 5.9).
La deuxième caractéristique vous permet de restaurer l'écran pendant que vous

exécutez les commandes P ou T (voir la commande FLASH, section 5.9). La
troisième caractéristique vous permet de restaurer l'écran du programme
temporairement (voyez la commande RS, section 5.9).

Si Soft-ICE ne semble pas suivre votre programme en mode graphique,essayez
de basculer WATCHV sur ON (voir la section 5.9 pour des détails).

11.4 Débugging en mémoire paginée

Un point d'arrêt domaine ou un point d'arrêt mémoire qui est positionné
dans une zone EMM mappable restera à cette adresse même si la page EMM est
mappée.

Quand vous débuggez des programmes EMM, la commande EMMMAP peut aussi être
très utile. Voir la section 5.6 pour plus d'informations.

Les commandes D, E, S, F, et C peuvent être employées pour voir ou modifier
nimporte quel gestionnaire de page EMM. La page ne doit pas être actuellement
mappée. La syntaxe de ces commandes est identique à celles des commandes
utilisées pour les pages non-EMM, à l'exception des suivantes:

* Dans les commandes D, E, S, et F, la partie adresse de la commande doit être précisée de la façon suivante:
  Hhandle# Ppage# décalage
  où handle est un nombre précisant quel gestionnaire EMM employer, page est un nombre précisant quelle page EMM employer, et décalage est un nombre de 0 à 4000H, précisant le décalage depuis le début de la page.
  Exemple:

  DB H1 P3 0

  Cette commande affichera les octets de la page 3 du gestionnaire 1, en commençant au décalage 0.

* La commande C doit être précisée de la façon suivante:

  C Hhandle# Ppage# offset1 L longueur offset2

  où handle et page sont les mêmes que ci-dessus. offset1 est un nombre de 0 à 4000H, précisant le décalage depuis le début de la page, où le premier bloc de données devant être comparées est situé. offset2 est un nombre de 0 à 4000H, précisant le décalage depuis le début de la page, où le deuxième bloc de données devant être comparées est situé.
  Exemple:

  C H2 P4 00 L10 1000

  Cette commande comparera les 10 premiers octets de mémoire situés au décalage 0 de la page 4 du gestionnaire 2 avec les 10 premiers octets de mémoire situés au décalage

1000 de la page 4 du gestionnaire 2.

Note:
Les occurences suivantes des commandes D, E, S, F, et C continueront à utiliser
le gestionnaire et la page précisés en dernier. Pour revenir à la mémoire
conventionnelle, utilisez l'une des commandes ci-dessus avec un segment précisé dans le champ adresse, par exemple:

```
D 0:0
```

11.5 Débugging en mémoire étendue

Les commandes D, E, S, F, et C peuvent être employées pour voir ou modifier
la mémoire étendue. La mémoire étendue réservée par Soft-ICE ne peut pas être
affichée. La syntaxe de ces commandes est similaire à celle des commandes
qui sont utilisées pour la mémoire conventionnelle:

* Dans les commandes D, E, S, et F commande,la portion adresse de la commande doit être précisée de la façon suivante:
  M mégaoctet adresse
  où mégaoctet est un nombre précisant combien de mégaoctets employer, et adresse précise l'adresse dans le mégaoctet spécifié.
  Exemple:
  DB M 2 0:0
  Cette commande affichera des octets depuis le début du méga-octet commençant à l'adresse linéaire 200000H.
* La commande C doit être précisée de la façon suivante:
  C M mégaoctet address1 L longueur address2
  où mégaoctet et address1 sont les mêmes que ci-dessus.
  address2 précise l'adresse dans le mégaoctet spécifié, où le deuxième bloc de données devant être comparées est situé.

  Exemple:

  C M 3 1000:2000 L10 3000:4000

  Cette commande comparera les 10 premiers octets de mémoire situés à 1000:2000 avec les 10 premiers octets de mémoire situés à 3000:4000.

Note:
Les utilisations suivantes des commandes D, E, S, F, et C continueront
à utiliser le dernier mégaoctet précisé. Pour revenir au mégaoctet 0
(mémoire conventionnelle), employez une des commandes ci-dessus avec 0
précisé comme mégaoctet, par exemple:

```
D M 0
```

CHAPITRE 12

Problèmes spécifiques de débugging

Soft-ICE peut être un outil puissant utilisé tout seul. Ce chapitre décrit des techniques pour débugger des composants système en utilisant Soft-ICE tout seul. Quand on utilise Soft-ICE tout seul, la commande ACTION doit être positionnée sur HERE.

12.1 Pilotes de périphériques

Le débugging de pilotes de périphériques nécessite un débugger qui ne fait pas appel au DOS. Soft-ICE peut être employée tout seul si votre débugger utilise le DOS.

Il y a deux méthodes pour débugger des pilotes de périphériques:

1. Employez la commande MAP pour trouver l'emplacement de votre pilote. Affichez le header du pilote pour trouver une stratégie ou l'interruption d'entrée. Positionner un point d'arrêt à l'entrée du pilote vous donnera le contrôle dans le pilote. La trace, ou des points d'arrêt, vous permettront de continuer le débugging. Le débugging du code d'initialization d'un pilote de périphérique nécessite de faire un reset du système avec la commande BOOT. Utilisez la technique énoncée ci-dessus pour positionner un point d'arrêt dans le code du pilote. La commande BOOT garde Soft-ICE et les points d'arrêts positionnés en mémoire.
2. La deuxième méthode nécessite de placer un code spécial dans votre pilote. Faites cela avec la commande 13HERE ON (voir la section 5.4). Placez un opcode INT 3 (CCH) dans votre pilote de périphérique au point où vous voulez prendre le contrôle. Quand l'INT 3 est exécutée, le contrôle est donné à Soft-ICE. Vous pouvez alors employer une commande RIP pour positionner le pointeur d'instruction autour du INT 3.

Si vous désirez débugger votre séquence d'initialisation, soyez certain que Soft-ICE est chargé dans CONFIG.SYS avant le pilote que vous essayez de débugger. Placez la commande 13HERE ON dans la chaîne INIT ddu fichier S-ICE.DAT. Avec cette méthode vous ne devez pas utiliser la commande BOOT.

Si vous êtes en train de débugger votre pilote de périphérique symboliquement ou avec le source vous devez charger le fichier de symboles et les

fichiers
source séparément du pilote de périphérique. Le fichier de symboles et les
fichiers source sont chargés avec le chargeur de programme de Soft-ICE:
LDR.EXE. Quand LDR.EXE est employé pour charger uniquement les symboles et les
sources vous devez l'employer sous cette forme:

```
LDR nom-fichier.SYM
```

L'extension du fichier symboles doit être précisée. Voir la section 7.4 pour
plus de détails sur LDR.EXE.

Après avoir chargés le fichier de symboles et les fichiers source avec LDR.EXE
vous devez entrer dans Soft-ICE et reloger les symboles par rapport au début
de votre pilote de périphérique. Les symboles sont relogés avec la commande
Soft-ICE SYMLOC. La syntaxe de la commande SYMLOC est:

```
SYMLOC segment
```

La valeur de segment est obtenue avec la commande MAP. Voir la description de
la commande SYMLOC pour plus de détails.

12.2 Chargeurs de boot

Le débugging des chargeurs de boot ou des programmes auto-bootables
nécessite l'utilisation de Soft-ICE tout seul. Vous devez tout d'abord démarrer
sous DOS et charger Soft-ICE. La méthode la plus facile pour débugger des
chargeurs de boot consiste à positionner un point d'arrêt à une adresse connue
dans le chargeur de boot, puis d'employer alors la commande BOOT pour faire
un reset du système. Soft-ICE est maintenu en mémoire avec les points d'arrêt
établis. Si une adresse connue est difficile à trouver un point d'arrêt
d'exécution peut être établi à 7C0:0H avant le
la commande BOOT. C'est l'adresse où la ROM BIOS charge le secteur de
démarrage en mémoire.

Une autre méthode nécessite de positionner le mode 13HERE sur ON (voir la
section 5.4). Placez un opcode INT 3 (CCH) dans votre programme au point où
vous voulez reprendre le contrôle. Quand l'INT 3 s'exécute, le contrôle est
donné à Soft-ICE.
Vous pouvez aussi employer le débugging symboles et source pour débugger un
chargeur de boot.
Voir la commande SYMLOC pour plus d'informations sur la façon de reloger vos
symboles et source vers le segment où votre chargeur de boot a été chargé.

12.3 Routines d'interruption

Soft-ICE permet des points d'arrêt et la trace dans les routines d'interruption du système (minuterie, clavier, etc.).
La trace et la pose de points d'arrêt dans les routines d'interruption du système est permis avec Soft-ICE. Vous pouvez tracer la routine d'interruption du clavier tandis que Soft-ICE utilise le clavier pour ses entrées. Dans la plupart des cas, Soft-ICE doit être employée tout seul quand quand vous débuggez les routines d'interruption du système. Pour positionner un point d'arrêt sur l'adresse d'une routine d'interruption système, utilisez une des méthodes suivantes:

1. Utilisez la commande affiche double:
   DD numéro-interruption * 4 L 1
   L'adresse affichée est l'adresse de la première instruction de la routine d'interruption système. Positionnez un point d'arrêt d'exécution sur cette adresse.
2. Utilisez la commande:
   BPINT numéro-interruption

12.4 Systèmes d'exploitation non-DOS

Les systèmes d'exploitation non-DOS en mode d'adressage réel peuvent être débuggés avec Soft-ICE. Si le système d'exploitation n'est pas vraiment compatible DOS vous devrez charger Soft-ICE sous DOS, et utiliser alors la commande BOOT pour lancer le système d'exploitation non-DOS. Suivre les instructions pour débugger les séquences de boot et les programmes auto-bootables comme expliqué dans la section 12.2.

Les commandes MAP et WARN peuvent ne pas fonctionner correctement sous un système d'exploitation non-DOS, mais les points d'arrêt et les commandes de débugging fonctionneront correctement.

Pour débugger avec des symboles ou un source vous devez charger les fichiers de symboles et les fichiers source tandis que vous êtes encore sous DOS ou dans le mode compatible DOS de votre système d'exploitation.

APPENDICE A

LISTE DES COMMANDES FONCTIONNELLES

APPENDICE B

LISTE ALPHABETIQUE DES COMMANDES

APPENDICE C

FONCTION DES SEQUENCES DE TOUCHES

Séquence	Description

Déplacement de la fenêtree Soft-ICE:

CTRL _	Monter la fenêtre d'une case
CTRL _	Descendre la fenêtre d'une case
CTRL ->	Déplacer la fenêtre d'une case à droite
CTRL <-	Déplacer la fenêtre d'une case à gauche

Modifications de la taille de la fenêtre Soft-ICE:

ALT _	Expansion de la fenêtre
ALT _	Contraction de la fenêtre

Edition de la ligne de Commande:

->	Déplacer le curseur à droite
<-	déplacer le curseur à gauche
INS	Basculer en mode insertion
DEL	Effacer le caractère courant
HOME	Déplacer le curseur au début de la ligne
END	Déplacer le curseur à la fin de la ligne
_	Affiche la commande précédente
_	Affiche la commande suivante
SHIFT _	Scrolle l'affichage une ligne vers le haut
SHIFT _	Scrolle l'affichage une ligne vers le bas
PAGE UP	Scrolle l'affichage une page vers le haut
PAGE DN	Scrolle l'affichage une page vers le bas
BKSP	Efface le caractère précédent
ESC	Annule la commande courante

APPENDICE D

Messages d'Erreur et Descriptions

Cet appendice liste et explique les messages d'erreur pouvant être générés par Soft-ICE.

A General Protection Violation Has Occurred. This is typically caused by a protected mode instruction.
CS:IP = XXXX:XXXX
Type 'C' to Continue
Type 'R' to Return to Soft-ICE.

The message arrive quand une instruction du mode protégé du 80386 est rencontrée ou s'il se produit un chevauchement de segments. Vous pourrez souvent déterminer la cause de ce message en désassemblant l'instruction à l'adresse spécifiée.
Si le premier octet de l'instruction est 0FH, il s'agit probablement d'une instruction du mode protégé. Si l'instruction tente d'accéder à un mot à l'offset 0FFFFH dans un segment, il s'agit alors d'un problème de chevauchement. Si vous entrez C pour continuer, le contrôle est passé au gestionnaire de l'interruption 0

dans la machine virtuelle DOS.

Ce message arrive souvent lorsqu'un programme exécute un saut vers une adresse ne contenant pas de code valide, ou dont le code a été altéré.

Attempt To Divide By 0

Ce message est affiché lorsque Soft-ICE évalue une expression et que le diviseur d'une division est égal à 0.

BPM Break Point Limit Exceeded

Soft-ICE autorise un maximum de 4 points d'arrêt mémoire. Ce message est affiché si vous tentez de dépasser cette limite.

Break Point Table Full

Soft-ICE autorise un maximum de 16 points d'arrêt. Ce message est affiché si vous tentez de dépasser cette limite.

Count Too Large

La commande Point d'Arrêt de Soft-ICE permet d'entrer une valeur optionnelle 'count'. Ce champ peut contenir une valeur maximum de FFH. Ce message d'erreur est affiché si la valeur spécifiée est supérieure à FFH.

DOS Memory Structures Corrupted

Ce message est affiché si Soft-ICE détecte un problème dans le chaînage des blocs mémoire DOS lors de l'utilisation de la commande MAP. Ce message peut aussi se produire si vous utilisez la commande MAP avec un système d'exploitation non-DOS.

Duplicate Break Point

Lorsqu'un point d'arrêt est entré, Soft-ICE compare les conditions de ce dernier aves celles des points d'arrêt précédemment entrés. S'il touve des correspondances, ce message est affiché.

Interrupt Break Point Limit Exceeded

Soft-ICE vous permet un maximum de 10 points d'arrêt interruption. Ce message est affiché si vous tentez de dépasser cette limite.

Invalid Opcode Has Occurred
CS:IP =XXXX:XXXX
Type 'C' to Continue
Type 'R' to Return to Soft-ICE.

Quand le 80386 rencontre une instruction illégale, il génère une interruption 6. Soft-ICE affiche ce message et vous donne la possibilité de continuer ou de retourner à Soft-ICE. Si vous choisissez C comme continuer, le contrôle est dooné au gestionnaire de l'interruption 6 dans la machine virtuelle.

Ce message arrive souvent lorsqu'un programme exécute un saut vers une adresse ne contenant pas de code valide, ou dont le code a été altéré.

I/O Break Point Limit Exceeded

Soft_ICE permet un maximum de 10 points d'arrêt. Ce message est affiché si vous tentez de dépasser cette limite.

No Alternate Screen

Ce message est affiché si la commande ALTSCR est utilisée et Soft-ICE ne détecte qu'une carte vidéo.

Parameter is Wrong Size

Certains champs nécessitent une taille de donnée particulière (octet, mot, double mot). Ce message est affiché si la taille de la donnée n'est pas corecte.
Par exemple, si vous utilisez la commande 'BPMP 2000:2000 EQ 1234', vous demandez à Soft-ICE de surveiller les accès octet à l'adresse 2000:2000 avec la valeur 1234H. Comme 1234H est plus grand qu'un octet, la commande génère le message d'erreur.

Parameters Required

Beaucoup de commandes de Soft-ICE nécessitent un ou plusieurs paramètres. Si une commande n'est pas entrée avec le nombre correct de paramètres, ce message est affiché.

Range Break Point Limit Exceeded

Soft-ICE permet unmaximum de 10 points d'arrêts sur un domaine mémoire. Ce message est affiché si vous tentez de dépasser cette limite.

Second Parameter Must Be Greater than First

Quand vous spécifiez un domaine mémoire, le premier nombre entré doit être la limite inférieure du domaine, sinon ce message est affiché.

Segment:Offset Can Not Wrap

La plupart des commandes de Soft-ICE ne permettent pas à un pointeur mémoire (segment:offset) de pointer d'abord en haut de mémoire puis en bas. Par exemple, le pointeur mémoire FFFF:FFFF enveloppe 2 zones de mémoire et est illégal. Ce message est affiché si vous tentez d'envelopper de la mémoire haute vers la basse.

Soft-ICE cannot be loaded.
Needs to load at top of memory.
Load before any TSR's or control programs.

Soft-ICE doit se charger lui-même le plus haut possible en mémoire. Cette mémoire est ensuite cachée à MS-DOS et rendue invisible pour les programmes qui ne peuvent ainsi pas détruire Soft-ICE. Ce message est affiché si Soft-ICE détecte qu'un autre programe s'est chargé avant

lui en haut de la mémoire.

Soft-ICE cannot run with other 80386 control programs

Le 80386 ne permet l'exécution que d'un seul programme en mode protégé à la fois, et donc, Soft-ICE ne peut pas cohabiter avec d'autres programmes de contrôle. En débuggant des programmes utilisant de l'EMS et de l'EEMS, vous pouvez obtenir ce message d'erreur quand vous essayez de charger Soft-ICE car certains systèmes 80386 sont livrés avec un programme de contrôle qui utilise le système de pagination du 80386 pour vous fournir de l'EMS et de l'EEMS avec une carte qui ne comporte que de la mémoire étendue. Vous pouvez, naturellement, utiliser une vraie carte de mémoire paginée pour débugger des programmes utilisant de l'EMS et de l'EEMS.

Soft-ICE has already been loaded

Ce message s'affiche si vous essayer de lancer une deuxième fois Soft-ICE.

Soft-ICE has not been loaded

Ce message s'affiche si vous essayez de décharger Soft-ICE alors qu'il n'a jamais été chargé.

Soft-ICE loads at the top of extended memory. This may conflict with other programs that use extended memory. If you are sure it will not conflict, then answer 'Y', otherwise answer 'N' and refer to the chapter on loading Soft-ICE with extended memory.

Ce message survient si vous tentez de charger Soft-ICE en mémoire étendue et que S-ICE.SYS n'a pas été chargé par votre fichier CONFIG.SYS. Cet avertissement sert à éviter que vous détruisiez par mégarde un Ram-Disk ou tout autre qui pourrait être chargé en mémoire étendue. Pour plus d'informations, référez-vous à la section 2.2, "Chargement de Soft-ICE" et au chapitre 6, "Options d'initialisation".

Soft-ICE will only run on 80386 based machines

Soft-ICE nécessite un microprocesseur 80386.

Syntax Error

Cemessage est affiché si les informations entrées sur la ligne de commande ne correspondent à la structure d'aucune commande.

The P & G Commands Function In RAM Only

Soft-ICE utilise deux méthodes pour implémenter les commandes P et G. La première méthode utilise les points d'arrêt registre du 80386. Ainsi, si vous avez déjà positionné 4 points d'arrêt style BPM, Soft-ICE utilise la méthode INT 3 qui ne fonctionne qu'en RAM. Si vous tentez d'utiliser les commandes P ou G en ROM maintenant, Soft-ICE détecte cette condition et affiche

ce message d'erreur.

Valid Verbs are R, W, RW, X

Ce message est affiché si un verbe incorrect est entré dans une commande BPM. les choix de verbes valides pour une commande BPM sont R(read), W(write),RW(read/write), and X (execute).

APPENDICE E

GUIDE DE DEPANNAGE

Cette appendice donne des solutions a de possibles problèmes que vous pouvez rencontrer lors de l'utilisation de Soft-ICE. Si vous ne trouvez
pas un problème ici, relisez le fichier README.SI sur votre disquette de distribution pour tous les trucs de dépannage ne figurant pas dans ce manuel.

L'heure n'est pas correcte à la fin de la journée.

Soft-ICE ne laisse passer aucune interruption vers le système lorsque sq fenêtre est affichée. Ceci n'affecte pas l'horloge temps réel et tout rentrera dans l'ordre au prochain reset de votre machine. Vous pouvez aussi corriger l'heure en lançant le programme UPTIME. Il prend l'heure dans l'horloge temps réel et appelle le DOS pour remettre l'heure à jour.

En débuggant un programme qui utilise de l'EMS et de l'EEMS, vous obtenez ce message d'erreur quand vous essayez de charger Soft-ICE "Soft-ICE cannot run with other 80386 control programs".

Certains systèmes 80386 sont livrés avec un programme de contrôle qui utilise le système de pagination du 80386 pour vous fournir de l'EMS et de l'EEMS avec une carte qui ne comporte que de la mémoire étendue. Le 80386 ne permet l'exécution que d'un seul programme en mode protégé à la fois, et donc, Soft-ICE ne peut pas cohabiter avec d'autres programmes de contrôle. Vous pouvez, naturellement, utiliser une vraie carte de mémoire paginée pour débugger des programmes utilisant de l'EMS et de l'EEMS.

Soft-ICE n'arrive pas à interrompre votre débugger.

Certains debuggers soft ne fonctionnent correctement qu'avec un seul type de debugging par interruption. Référez-vous à la commande ACTION à la section 5.4. Il y est listé trois types d'actions standard qui peuvent être choisis lors de l'exécution d'un point d'arrêt.

Essayez les trois. Certaines marchent mieux que d'autres avec certains debuggers.

Soft-ICE ne fonctionne pas correctement quand votre moniteur est en mode graphique, ou il ne restaure pas correctement l'écran graphique.

Soft-ICE n'utilise pas la ROM BIOS pour ses sorties. Il écrit directement dans la mémoire. Soft-ICE a été prévu pour fonctionner avec les types de contrôleurs suivants ou des modèles 100% compatibles:

- CGA
- MDA
- Hercules
- EGA
- VGA

Si voitre contrôleur n'est pas parmi ceux-là, ou n'est pas 100% compatible, vous devrez ytiliser un second contrôleur et moniteur et utiliser la

commande ALTSCR décrite à la section 5.9.

La séquence de touches utilisée pour activer Soft-ICE entre en conflit
avec un de vos autres programmes.

Vous pouvez modifier la séquence de touches utilisée pour activer Soft-ICE à l'aide de la commande ALTKEY. référez-vous à la commande ALTKEY à la section 5.8.

Quand votre programme plante, Soft-ICE n'est plus accessible.

Référez-vous à la commande BREAK à la section 5.4. Cette commande vous permet d'activer la fenêtre Soft-ICE quand le système est planté avec les interruptions désactivées.

Après qu'un point d'arrêt ait activé votre débugger, celui-ci ne répond plus.

Il y a deux raisons possibles à celà:

1) Votre débugger a causé une ré-entrance au niveau du DOS ou de la ROM BIOS. Le DOS et LE BIOS n'étant pas pleinemant ré-entrants, votre débugger ne peut pas fonctionner correctement. Utiliser la commande WARN pour positionner l'avertissement de ré-entrance sur ON. La prochaine fois que le DOS sera prêt à ré-entrer un message d'avertissement sera affiché et vous pourrez choisir de retourner dans Soft-ICE pour éviter le problème. Référez-vous à la commande WARN à la section 5.4.
2) Un break s'est produit au milieu d'une routine d'interruption. Certains débuggers ne savent pas traiter ce problème. Utilisez ACTION positionné sur HERE, car Soft-ICE vous permet les breaks au milieu des routines d'interruption. Référez-vous à la section 5.4.

Vous utilisez un moniteur CGA et il y a beaucoup de neige quand Soft-ICE est activé.

Certains types de cartes video affichent de la neige si les caractères sont affichés sans attendre un balayage horizontal ou vertical. Pour réduire la neige, basculez le mode FLICK ON. Référez-vous à la commande FLICK à la section 5.9.

Quand vous utilisez la commande BOOT, le système commence un redémarrage, puis se plante.

Soft-ICE utilise l'interruption 19 pour booter à chaud. Il y a deux causes pour lesquelles ceci pourrait ne pas marcher:

1) Sur un système redémarré depuis peu, cette méthode fonctionne bien. Mais si un programme "errant" a endommagé le système, cela peut échouer.
2) Certains programmes utilisant de la mémoire étendue ou paginée comme les pilotes EMS ou les caches disques ne sont pas capables de reconnaître un reset causé par l'interruption 19. Quand vous débuggez des pilotes de périphériques ou des

chargeurs de boot qui ont ce problème, vous devez utiliser la méthode suivante. Démarrez le système sans les pilotes causant problème; Chargez Soft-ICE; préparez les pilotes pour chargement au prochain démarrage; et enfin utilisez la commande BOOT. Référez-vous à la commande BOOT à la section 5.3.

Vous venez juste d'utiliser la touche SYSREQ pour activer Soft-ICE et votre système semble bloqué, ou il a commancé à faire une copie d'écran sur votre imprimante.

Sur certains claviers, pour générer un System Request, il faut appuyer simultanément sur les touches ALT et PrtScr. Si, accidentellement, vous n'appuyez que sur la touche PrtScr, le système tentera de faire une copie d'écran sur votre imprimante. Si vous n'avez pas d'imprimante connectée, le système semblera bloqué. Pour éviter ce problème, appuyez avec attention sur les deux touches ou utilisez la commande ALTKEY pour définir une séquence de touches différente. Référez-vous à la commande ALTKEY à la section 5.8.

Vous désassembliez des instructions, ou éditiez ou affichiez la mémoire quan votre débugger s'est bloqué.

Vous avez accédé une adresse qui a déclenché un point d'arrêt de Soft-ICE dont l'ACTION n'était pas positionné sur HERE. Quand Soft-ICE arrive au point ou vous souhaitez examiner la mémoire avec votre débugger, vous devez désarmer les point d'arrêt de Soft-ICE. Si vous ne le faites pas, vous risquez de reprovoquer un point d'arrêt. Ceci va ammener votre débugger à s'activer lui-même ce qui peut être fatal s'il n'est pas ré-entrant.

Après votre sortie de votre débugger, votre système se bloque.

Ce problème peut naturellement avoir plusieurs causes. Un parmi celles possibles est que vous ayez oublié de désactiver les points d'arrêt de Soft-ICE et ACTION est toujours postionné pour activer votre débugger. Quand le point d'arrêt se produit, ACTION tente d'activer votre débugger mais celui-ci n'est plus en mémoire.

Vous positionnez un point d'arrêt pour piéger l'interrution 15H, fonctions 87H, 88H, ou 89H et le point d'arrêt ne s'exécute pas.

Soft-ICE traite ces fonctions de façon interne en mode protégé. Il n'est donc pas possible d'y poser des points d'arrêt.

Votre programme n'accepte plus d'entrée clavier, mais le clavier est toujours actif.

Une touche shift est peut-être bloquée logiquement basse. Essayer d'appuyer puis de relacher chaque shift, control et alt.

Soft-ICE ne peut pas restaurer l'affichage correctement.

Soft-ICE a une émulation vidéo puissante qui lui permet de simuler beaucoup de modes graphiques spéciaux. Activez ce mode par la commande WATCHV ON. Regardez la

description de la commande WATCHV pour plus de détails. Pour des contrôleurs vidéo non compatibles et pour certains modes rares, il vous faudra utiliser un deuxième moniteur. Voir la commande ALTSCR.

La machine se bloque quand vous êtes dans Soft-ICE ou quand vous déplacez la fenêtre de Soft-ICE.

Soft-ICE a des problèmes de timing avec certains claviers. Utilisez la commande NOLEDS dans S-ICE.DAT. Ceci empêche Soft-ICE d'envoyer des commandes aux LED's du clavier.